Loan Prediction Analysis

॥वसुधैव कुटुम्बकम्॥

PROJECT REPORT SUBMITTED TO
Symbiosis Institute of Geoinformatics

FOR PARTIAL FULFILLMENT OF THE M. Sc. DEGREE

By
PRANAV KATARIYA

PRN 22070243026
M.Sc. (Data Science and Spatial Analytics)

BATCH 2022-24

Symbiosis Institute of Geoinformatics

Symbiosis International (Deemed University)
5th Floor, Atur Centre
Gokhale Cross Road
Model Colony
Pune – 411016
Maharashtra
India

February 2023

Acknowledgment

I would like to convey my heartfelt gratitude to Dr. Vidya Patkar and Mr.Sahil Shah for their tremendous support and assistance in completing my project. I would also like to thank them, for providing me with this beautiful opportunity to work on a project with the topic “**Loan Prediction Analysis**”. The completion of the project would not have been possible without their help and insights.

I am grateful to all those with whom I have had the pleasure to work during this and other related projects. Nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents, whose love and guidance are with me in whatever I pursue. They are the ultimate role models.

Pranav Katariya

INDEX

Abstract:

This report presents a loan prediction analysis that aims to predict the likelihood of loan approval for customers based on various factors such as credit history, income, loan amount, and others. The analysis was conducted using a dataset of customer information and loan details from a company named Dream Housing Finance. Different machine learning algorithms were employed, including SVM, Naïve Bayes, logistic regression, decision tree, and random forest, to evaluate their effectiveness in predicting loan approval. The results indicate that Random Forest algorithm performed the best with an accuracy of 79%, followed by the Logistic Regression and the Naïve Bayes algorithm with an accuracy of 77%, the Decision Tree with an accuracy of 70% and the SVM with the lowest accuracy of 65%. The analysis also revealed that credit history was the most significant factor in determining loan approval. Based on these findings, the report concludes with recommendations for the financial institution to improve its loan approval process and reduce the risk of default.

Introduction :

The banking and financial industry plays a significant role in the growth and development of any economy. Banks and other financial institutions provide various types of loans to individuals and businesses to finance their operations, investments, and other financial needs. However, the risk of default on these loans is a major concern for lenders. Predicting the likelihood of loan defaults can help lenders mitigate this risk and make more informed lending decisions.

This report presents the results of a loan prediction analysis conducted on a dataset of historical loan applications. The objective of this analysis is to develop a predictive model that can accurately identify loan applicants who are likely to default on their loans. The analysis uses a range of statistical and machine learning techniques to identify the most important factors that contribute to loan defaults and to develop a predictive model that can be used to assess the risk of loan applicants.

The report is organized as follows: the next section provides a brief overview of the dataset used in the analysis, including the variables and the sample size. The third section presents the exploratory data analysis (EDA) performed on the dataset, which aims to identify any patterns or relationships between the variables. The fourth section describes the analytical techniques used to develop the predictive model, including feature engineering, model selection, and evaluation. The fifth section presents the results of the analysis, including the performance metrics of the predictive model and the key factors that contribute to loan defaults. The report concludes with a summary of the main findings and recommendations for future action.

Data Sources :

The dataset used in this analysis is taken from the **Kaggle**. It is an online community and platform for data scientists and machine learning enthusiasts. It provides a wide range of publicly available datasets that can be used for various data analysis and machine learning tasks. These datasets are contributed by the Kaggle community or by organizations and individuals who want to share their data with others for analysis.

To view the dataset click here.

Database system used:

PostgreSQL –It is an open source object-relational database management system (ORDBMS) developed by a worldwide team of volunteers. It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users. It is the most advanced open source database available, with features such as Multi-Version Concurrency Control (MVCC), point in time recovery, tablespaces, asynchronous replication, nested transactions (savepoints), online/hot backups.

Uniqueness:

"Loan Prediction Analysis" is a unique task that involves predicting whether a loan applicant is likely to be approved or denied based on various factors such as their credit score, income, and employment history. It is a critical task for financial institutions and lenders to manage risks and make informed decisions about loan approvals. Additionally, the task involves handling large volumes of data, dealing with missing values, and selecting appropriate machine learning models to make accurate predictions.

About dataset:

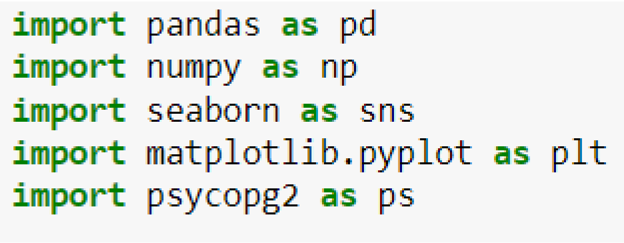
Dream Housing Finance company deals in all home loans. They have presence across all urban, semi urban and rural areas. Customer first apply for home loan after that company validates the customer eligibility for loan. Company wants to automate the loan eligibility process based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History, etc. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can target these customers. This is a standard supervised classification task. A classification problem where we have to predict whether a loan would be approved or not.

Below is the dataset attributes with description.

Variable	**Description**
Loan_ID	Unique Loan ID
Gender	Male/Female
Married	Applicant married(Y/N)
Dependents	Number of dependents
Education	Applicant Education(Graduate/Under Graduate)
Self_Employed	Self employed(Y/N)
ApplicantIncome	Applicant income
CoapplicantIncome	Coapplicant income
LoanAmount	Loan amount in thousands
Loan_Amount_Term	Term of loan in months
Credit_History	Credit history meets guidelines
Property_Area	Urban/Semi Urban/Rural
Loan_Status	Loan approved(Y/N)

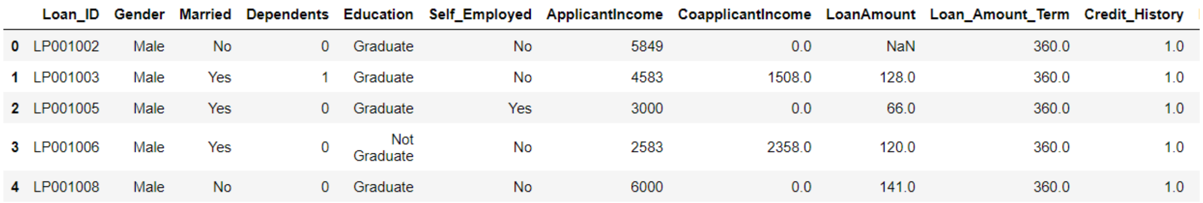
Data cleaning and Data Pre-processing:

Different Python libraries such as Pandas, NumPy, matplotlib ,psycopg2 , seaborn were used for the prediction and analysis of this report.

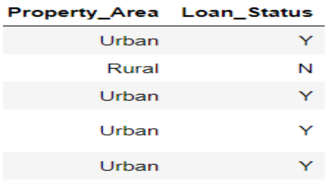


```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import psycopg2 as ps
```

Overview of the dataset-

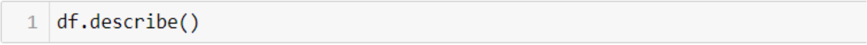


	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0

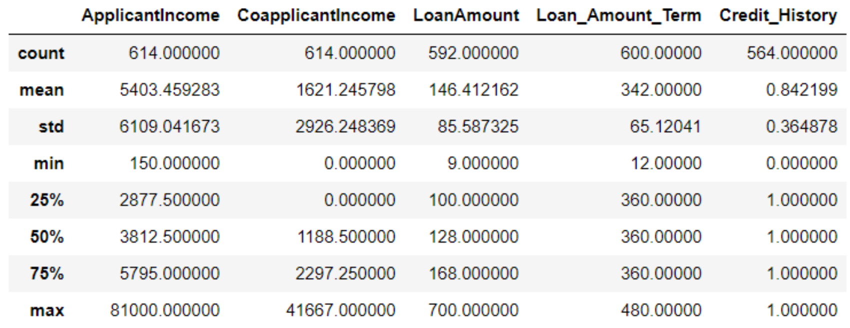


Property_Area	Loan_Status
Urban	Y
Rural	N
Urban	Y
Urban	Y
Urban	Y

Data description-

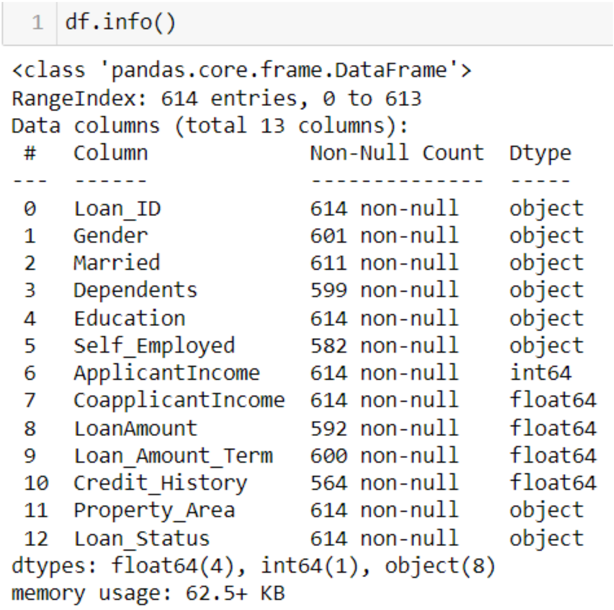


```python
df.describe()
```



	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

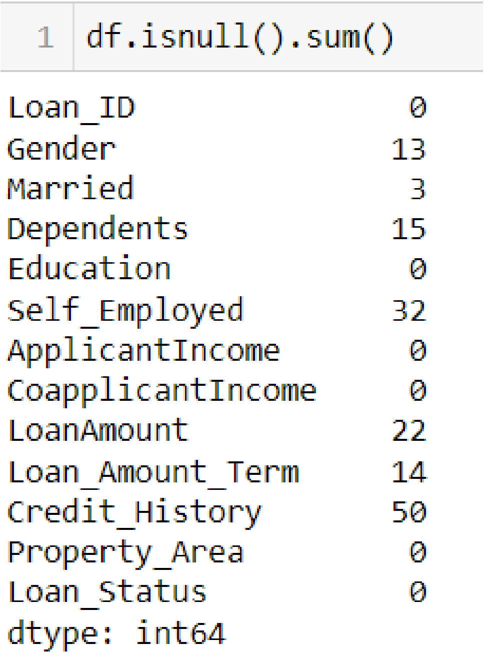
Dataset Information-



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            614 non-null    object 
 1   Gender             601 non-null    object 
 2   Married            611 non-null    object 
 3   Dependents         599 non-null    object 
 4   Education          614 non-null    object 
 5   Self_Employed      582 non-null    object 
 6   ApplicantIncome    614 non-null    int64  
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object 
 12  Loan_Status        614 non-null    object 
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

Preprocessing the dataset –



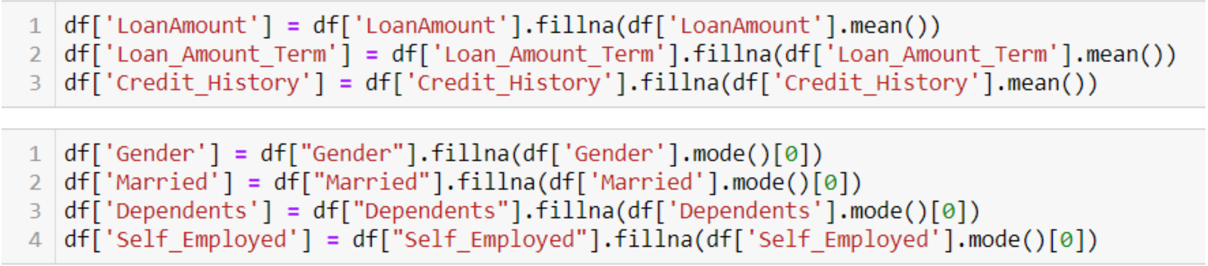
```
df.isnull().sum()
```

```
Loan_ID               0
Gender               13
Married               3
Dependents           15
Education             0
Self_Employed        32
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History       50
Property_Area         0
Loan_Status           0
dtype: int64
```

Filling the null values-

For numerical terms-mean

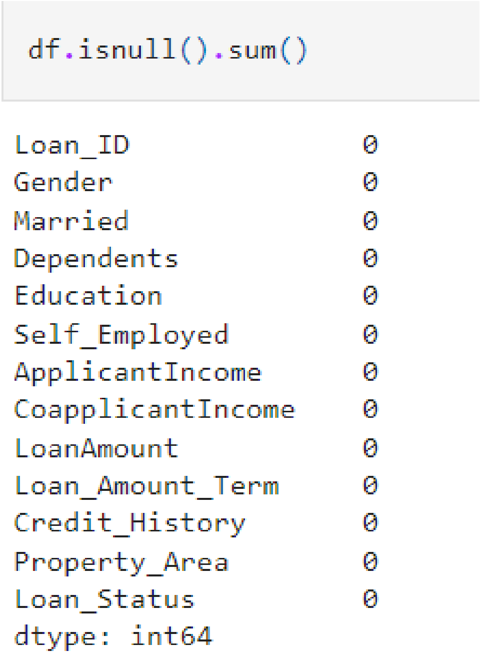
For categorical terms-mode



```python
df['LoanAmount'] = df['LoanAmount'].fillna(df['LoanAmount'].mean())
df['Loan_Amount_Term'] = df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mean())
df['Credit_History'] = df['Credit_History'].fillna(df['Credit_History'].mean())
```

```python
df['Gender'] = df["Gender"].fillna(df['Gender'].mode()[0])
df['Married'] = df["Married"].fillna(df['Married'].mode()[0])
df['Dependents'] = df["Dependents"].fillna(df['Dependents'].mode()[0])
df['Self_Employed'] = df["Self_Employed"].fillna(df['Self_Employed'].mode()[0])
```

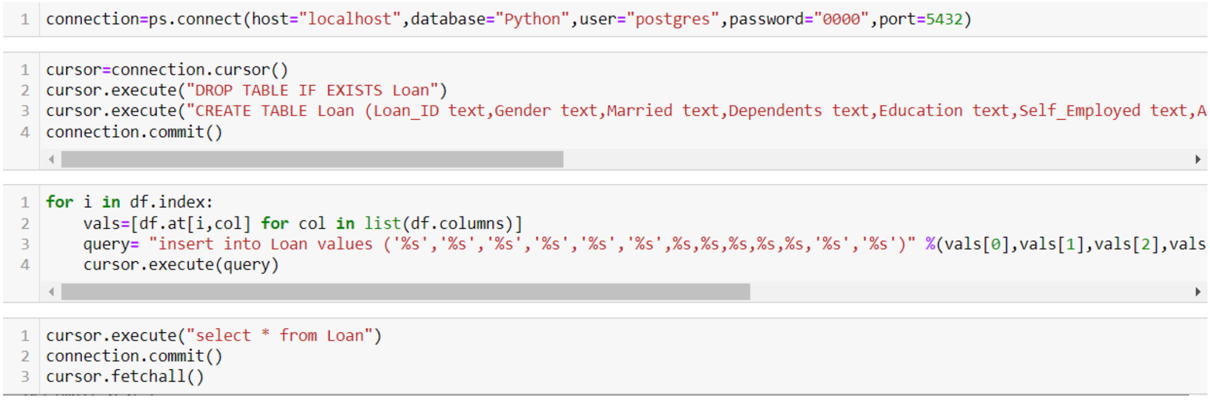
After Preprocessing-



```python
df.isnull().sum()
```

```
Loan_ID              0
Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
Loan_Status          0
dtype: int64
```

Connecting dataset to the database:

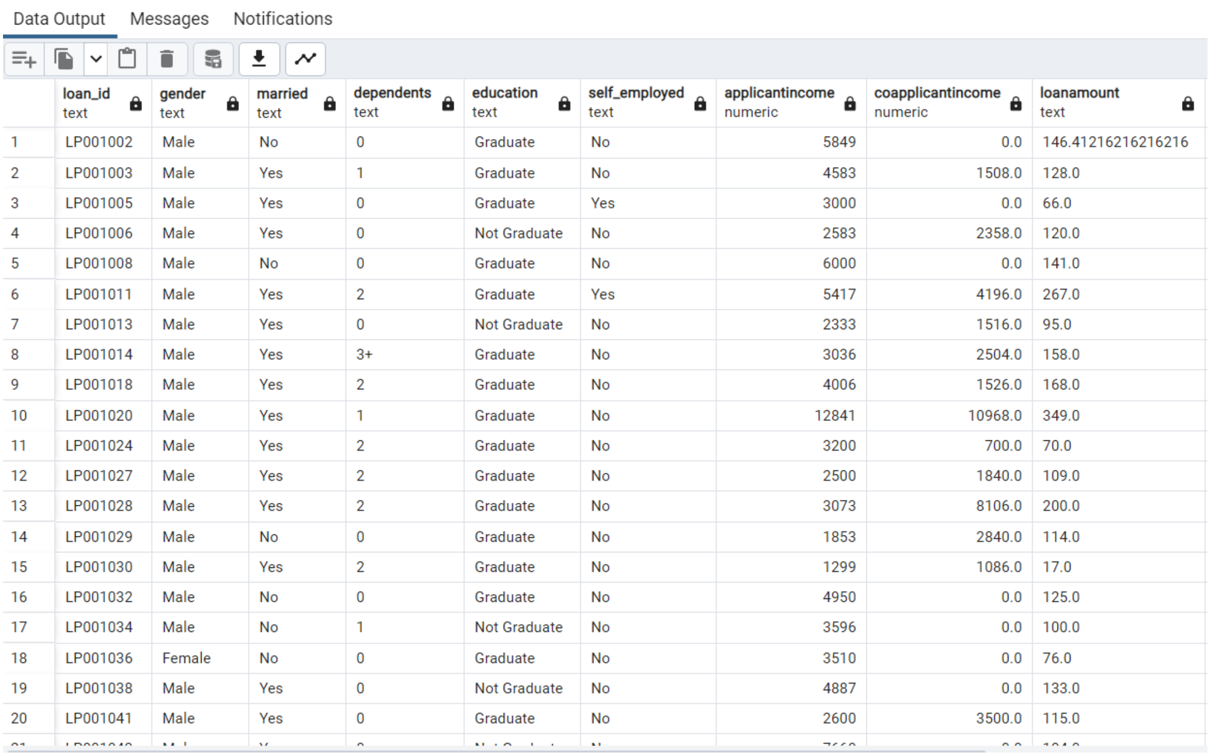


```python
connection=ps.connect(host="localhost",database="Python",user="postgres",password="0000",port=5432)
```

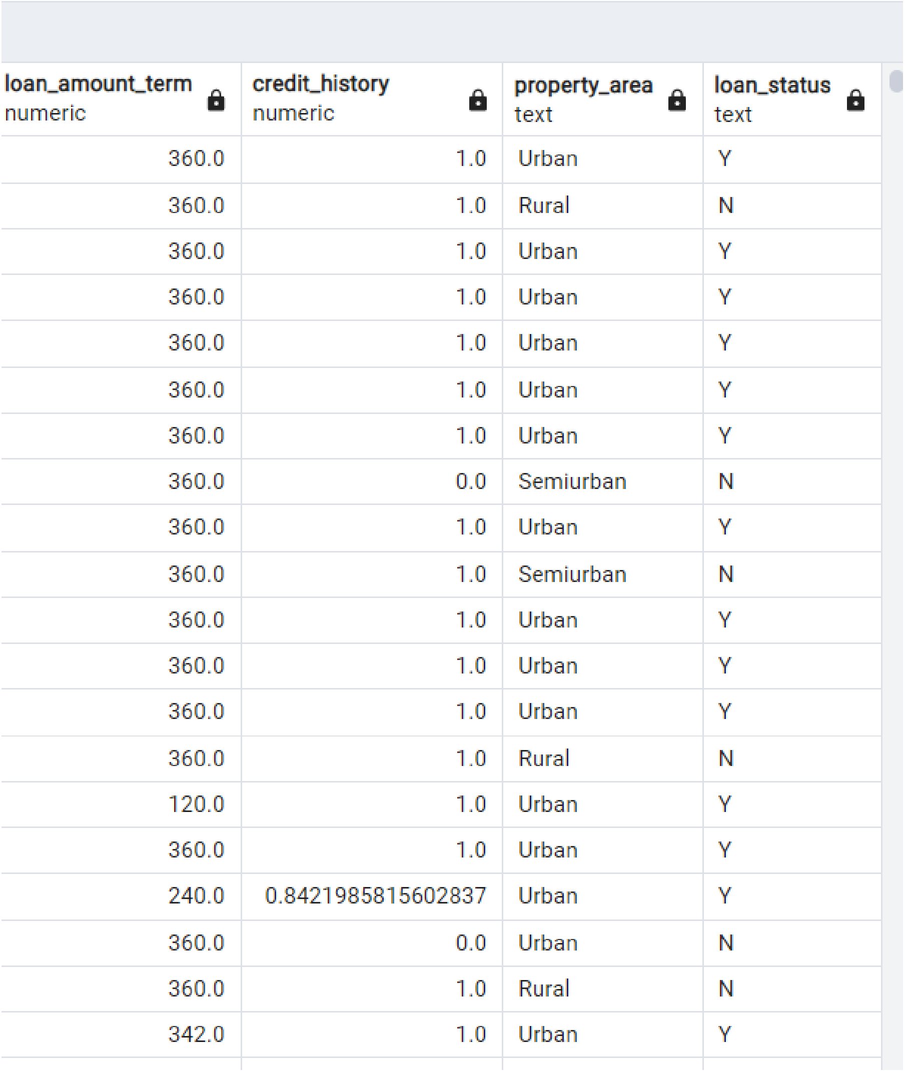
```python
cursor=connection.cursor()
cursor.execute("DROP TABLE IF EXISTS Loan")
cursor.execute("CREATE TABLE Loan (Loan_ID text,Gender text,Married text,Dependents text,Education text,Self_Employed text,A
connection.commit()
```

```python
for i in df.index:
    vals=[df.at[i,col] for col in list(df.columns)]
    query= "insert into Loan values ('%s','%s','%s','%s','%s','%s',%s,%s,%s,%s,%s,'%s','%s')" %(vals[0],vals[1],vals[2],vals
    cursor.execute(query)
```

```python
cursor.execute("select * from Loan")
connection.commit()
cursor.fetchall()
```



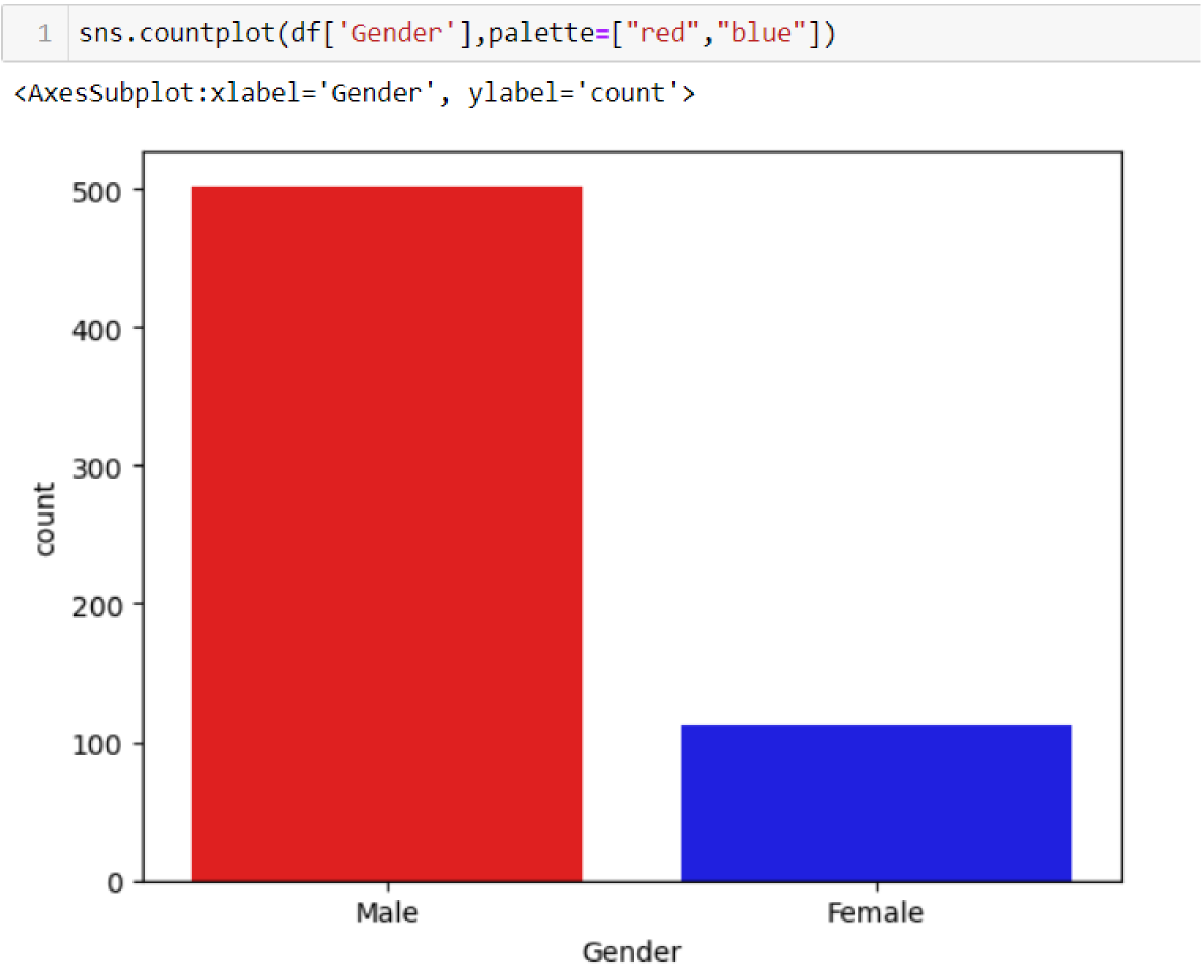
	loan_id text	gender text	married text	dependents text	education text	self_employed text	applicantincome numeric	coapplicantincome numeric	loanamount text
1	LP001002	Male	No	0	Graduate	No	5849	0.0	146.41216216216216
2	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0
3	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0
4	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0
5	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0
6	LP001011	Male	Yes	2	Graduate	Yes	5417	4196.0	267.0
7	LP001013	Male	Yes	0	Not Graduate	No	2333	1516.0	95.0
8	LP001014	Male	Yes	3+	Graduate	No	3036	2504.0	158.0
9	LP001018	Male	Yes	2	Graduate	No	4006	1526.0	168.0
10	LP001020	Male	Yes	1	Graduate	No	12841	10968.0	349.0
11	LP001024	Male	Yes	2	Graduate	No	3200	700.0	70.0
12	LP001027	Male	Yes	2	Graduate	No	2500	1840.0	109.0
13	LP001028	Male	Yes	2	Graduate	No	3073	8106.0	200.0
14	LP001029	Male	No	0	Graduate	No	1853	2840.0	114.0
15	LP001030	Male	Yes	2	Graduate	No	1299	1086.0	17.0
16	LP001032	Male	No	0	Graduate	No	4950	0.0	125.0
17	LP001034	Male	No	1	Not Graduate	No	3596	0.0	100.0
18	LP001036	Female	No	0	Graduate	No	3510	0.0	76.0
19	LP001038	Male	Yes	0	Not Graduate	No	4887	0.0	133.0
20	LP001041	Male	Yes	0	Graduate	No	2600	3500.0	115.0



loan_amount_term numeric	credit_history numeric	property_area text	loan_status text
360.0	1.0	Urban	Y
360.0	1.0	Rural	N
360.0	1.0	Urban	Y
360.0	1.0	Urban	Y
360.0	1.0	Urban	Y
360.0	1.0	Urban	Y
360.0	1.0	Urban	Y
360.0	0.0	Semiurban	N
360.0	1.0	Urban	Y
360.0	1.0	Semiurban	N
360.0	1.0	Urban	Y
360.0	1.0	Urban	Y
360.0	1.0	Urban	Y
360.0	1.0	Rural	N
120.0	1.0	Urban	Y
360.0	1.0	Urban	Y
240.0	0.8421985815602837	Urban	Y
360.0	0.0	Urban	N
360.0	1.0	Rural	N
342.0	1.0	Urban	Y

Data Analysis and Visualization:

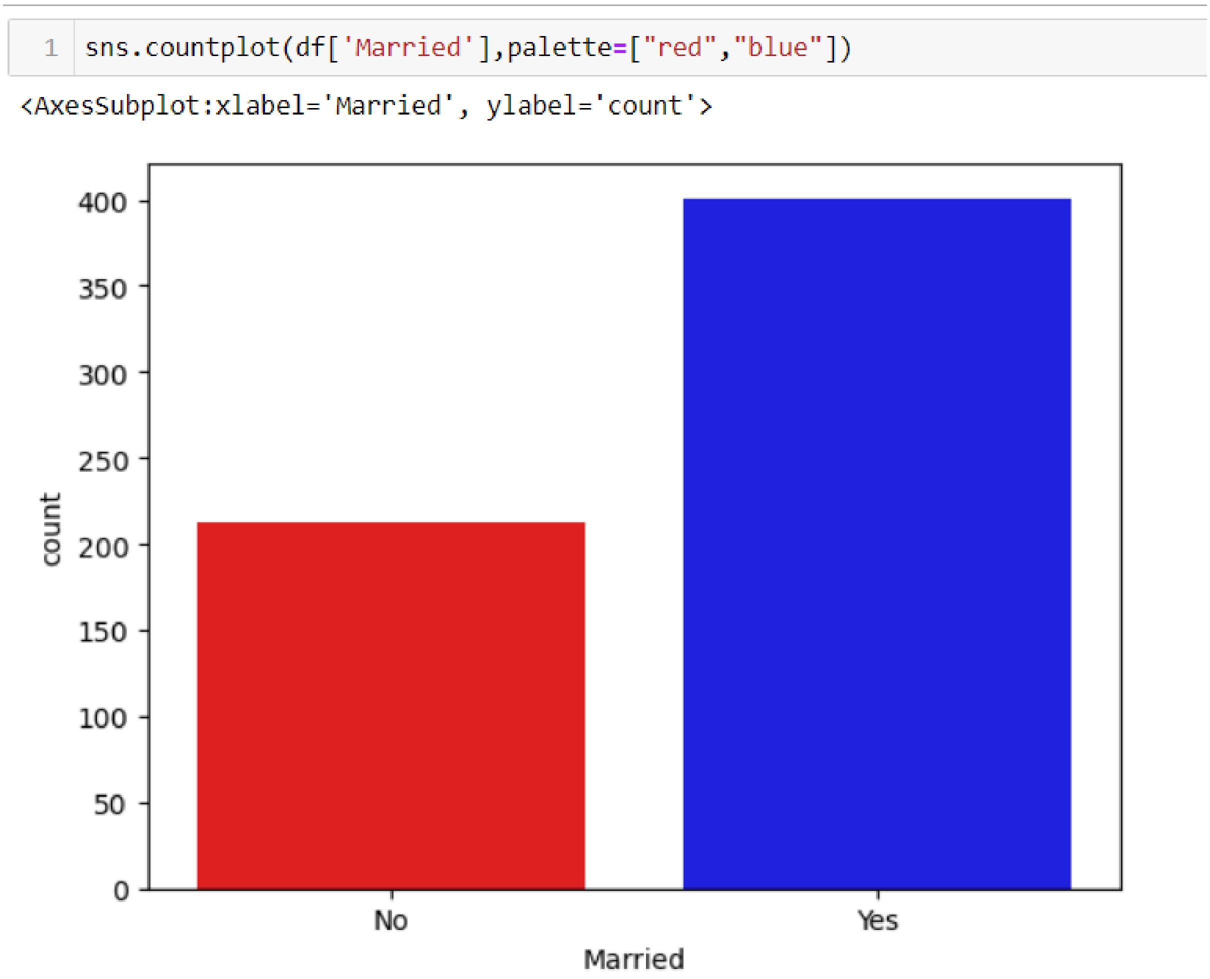
Exploratory Data Analysis:



```
sns.countplot(df['Gender'],palette=["red","blue"])
```

```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```

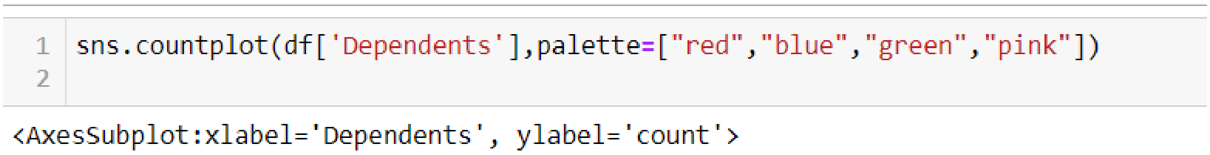
The countplot here shows that there is a higher proportion of male applicants than female applicants for loans with approx 500 applicants. This suggests that there may be a gender gap in access to financial services, with men being more likely to apply for loans than women. This could be due to a variety of factors, such as differences in financial literacy, access to credit, or cultural norms.



```
sns.countplot(df['Married'],palette=["red","blue"])
```

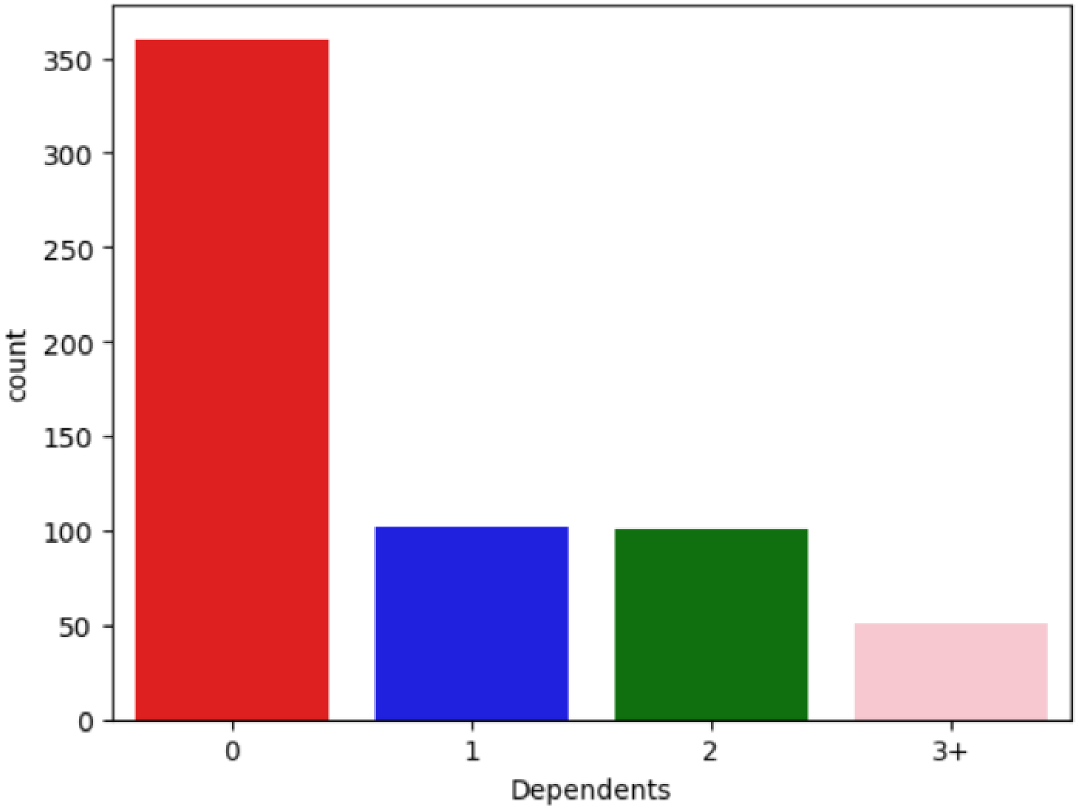
```
<AxesSubplot:xlabel='Married', ylabel='count'>
```

From the countplot, we can see that the majority of applicants for loans are married. This suggests that married individuals are more likely to apply for loans than those who are not married. This could be due to the fact that married couples may have more financial stability and resources to draw upon when applying for a loan, as well as the fact that they may have a better credit score due to their combined incomes.



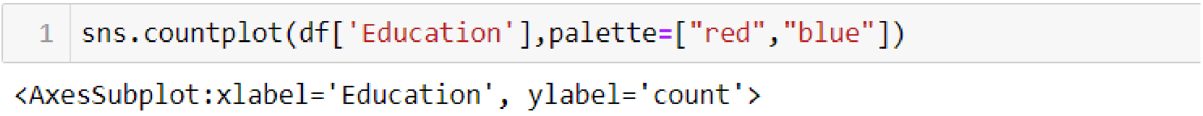
```
sns.countplot(df['Dependents'],palette=["red","blue","green","pink"])

```

```
<AxesSubplot:xlabel='Dependents', ylabel='count'>
```



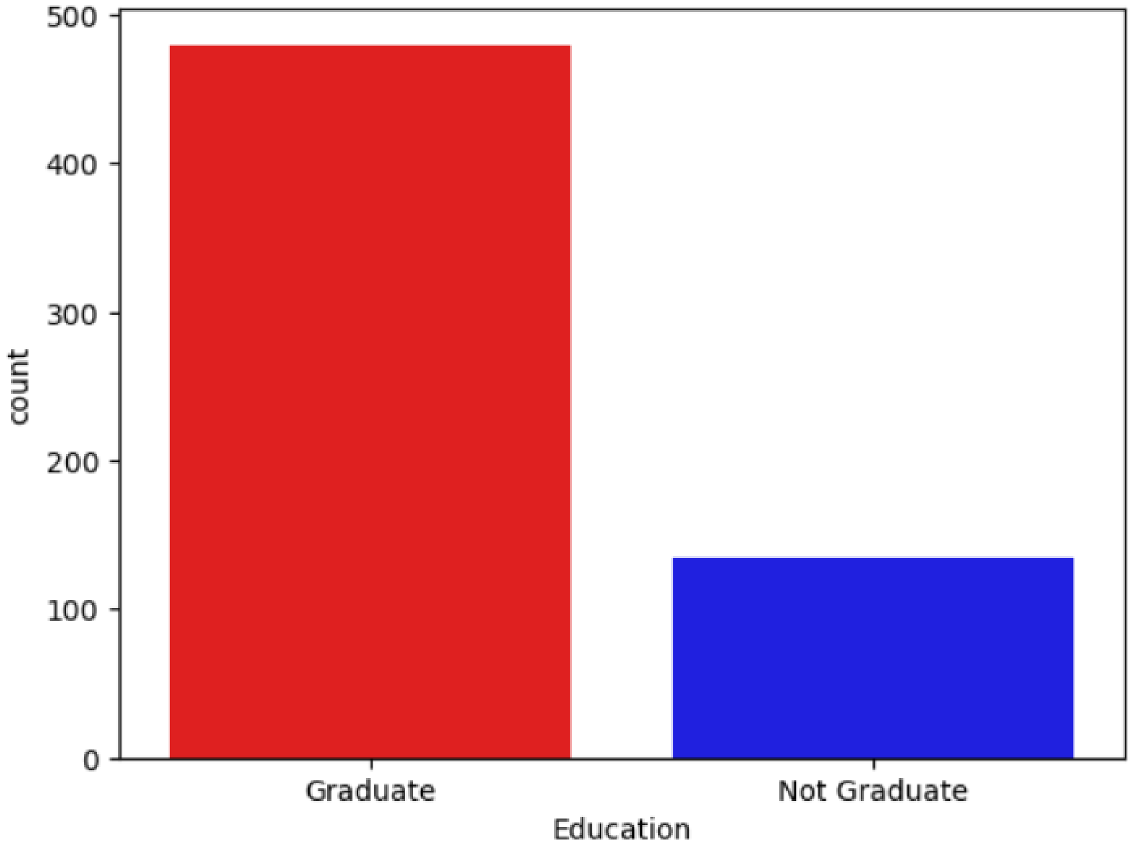
Dependents in loan are people who are financially dependent on the borrower, such as a spouse, children, or other family members. Lenders may consider the income and expenses of dependents when evaluating a loan application. This is because the borrower's ability to repay the loan may be affected by the financial obligations of their dependents.

The above countplot shows that the majority of loan applicants (over 50%) did not have any dependents. This suggests that most loan applicants were either single or did not have any dependents that would affect their ability to repay the loan. This could be due to the fact that lenders are more likely to approve loans to applicants who have a steady income and are less likely to be affected by financial obligations to dependents.

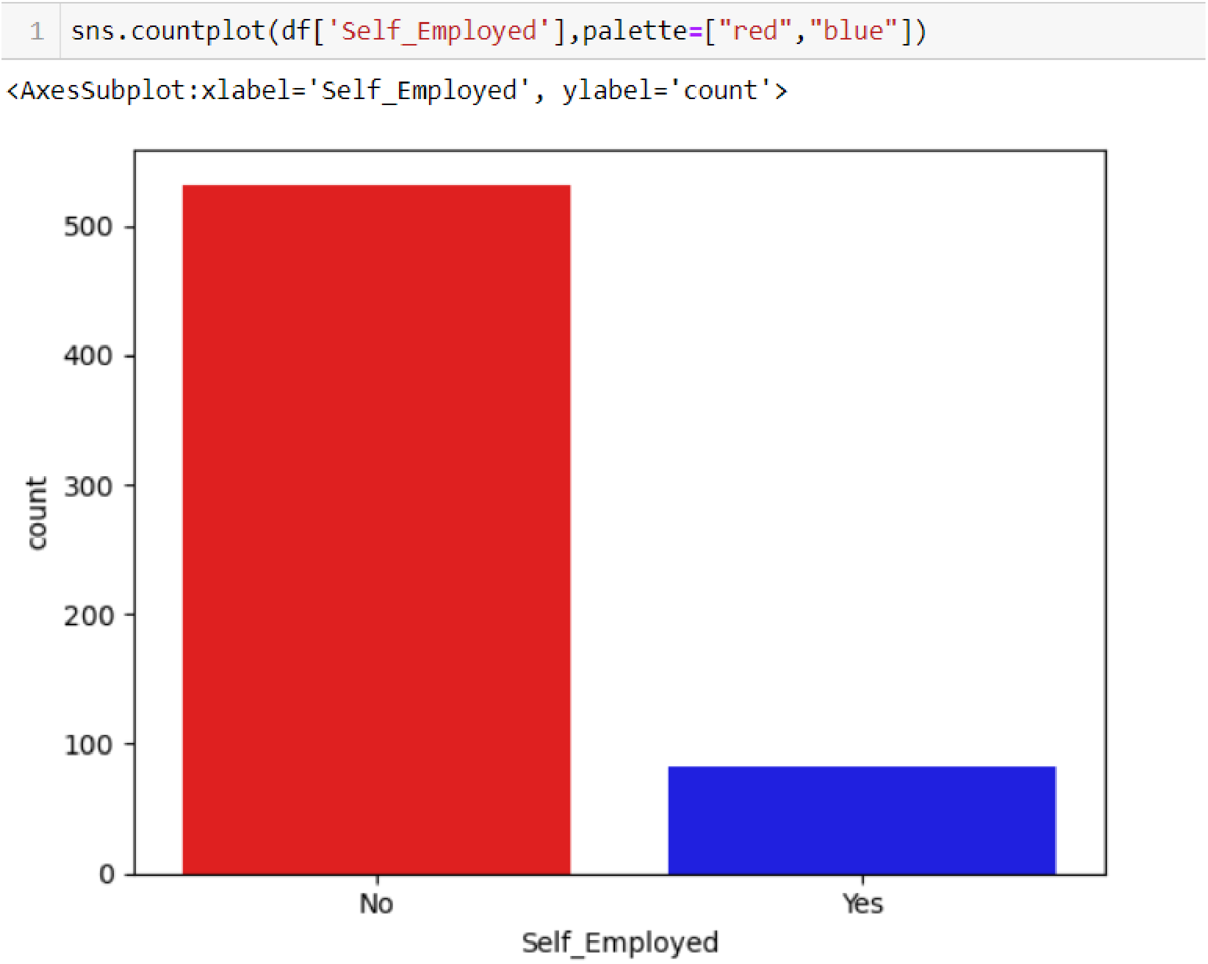


```
sns.countplot(df['Education'],palette=["red","blue"])
```

```
<AxesSubplot:xlabel='Education', ylabel='count'>
```



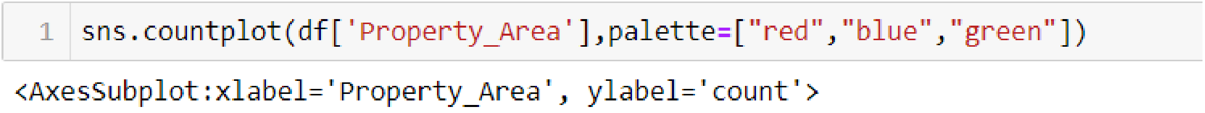
From the countplot we can see that the majority of people who apply for loans are those who have graduated from college or university. This suggests that having a higher level of education is associated with a higher likelihood of applying for a loan. This could be due to the fact that those with higher levels of education may have better access to financial resources and may be more likely to have the necessary qualifications to be approved for a loan.



```
sns.countplot(df['Self_Employed'],palette=["red","blue"])
```

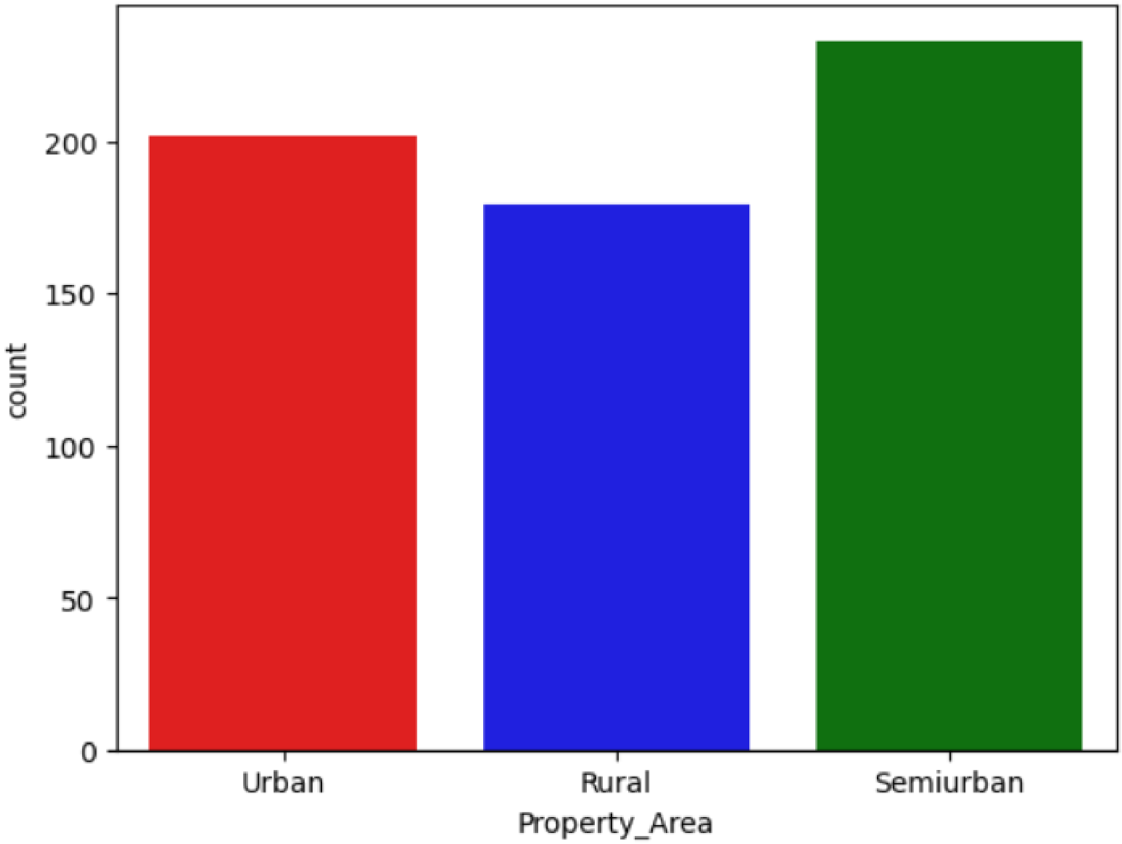
```
<AxesSubplot:xlabel='Self_Employed', ylabel='count'>
```

From the countplot, we can see that the majority of loan applicants are not self-employed. This suggests that self-employed individuals are less likely to apply for loans than those who are employed by an employer. This could be due to a variety of factors, such as the difficulty of proving income or the lack of access to credit for self-employed individuals. Additionally, self-employed individuals may be more likely to have irregular income, which could make it more difficult to qualify for a loan.

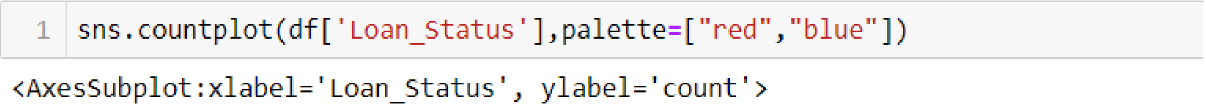


```
sns.countplot(df['Property_Area'],palette=["red","blue","green"])
```

```
<AxesSubplot:xlabel='Property_Area', ylabel='count'>
```

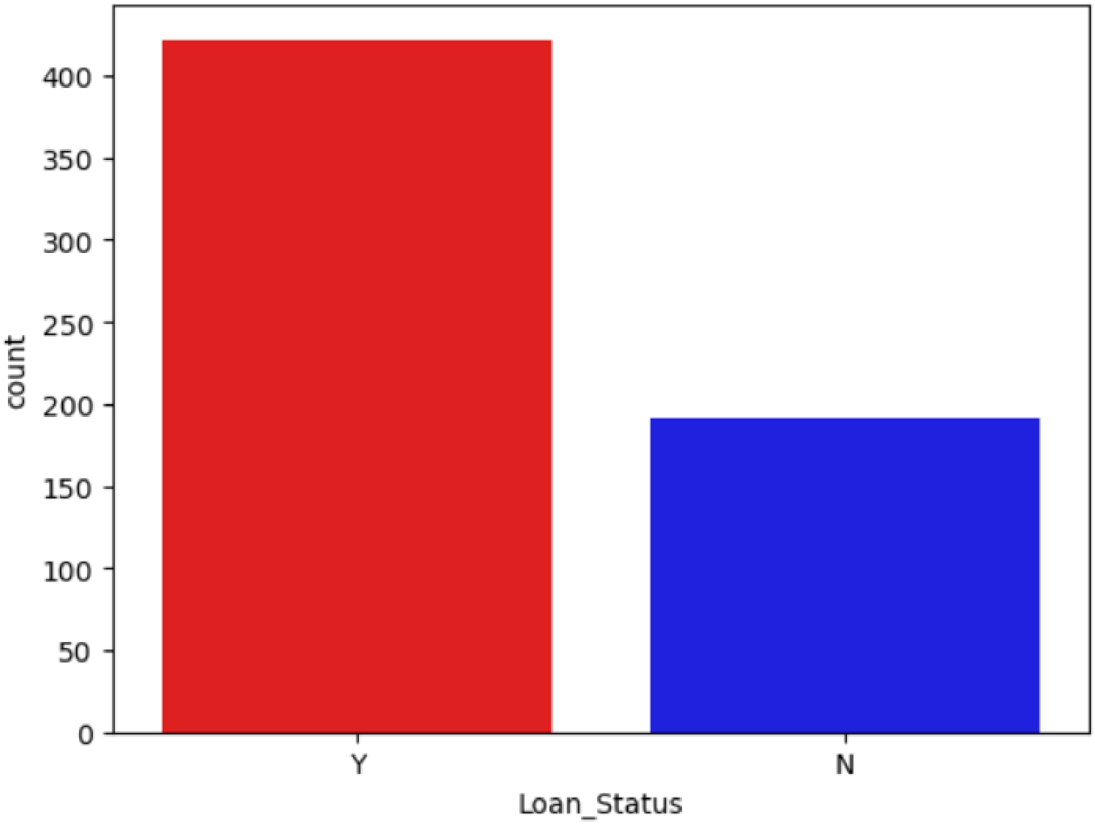


From the countplot, we can see that the majority of people who apply for loans have their property located in semi-urban areas. This is followed by urban areas and then rural areas. This suggests that people in semi-urban areas are more likely to apply for loans than those in urban or rural areas. This could be due to the fact that semi-urban areas have a higher population density than rural areas, and may have more access to financial services than rural areas.



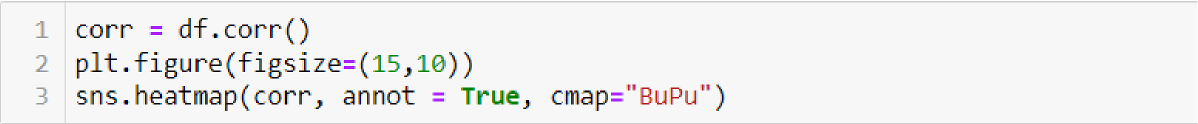
```
sns.countplot(df['Loan_Status'],palette=["red","blue"])
```

```
<AxesSubplot:xlabel='Loan_Status', ylabel='count'>
```



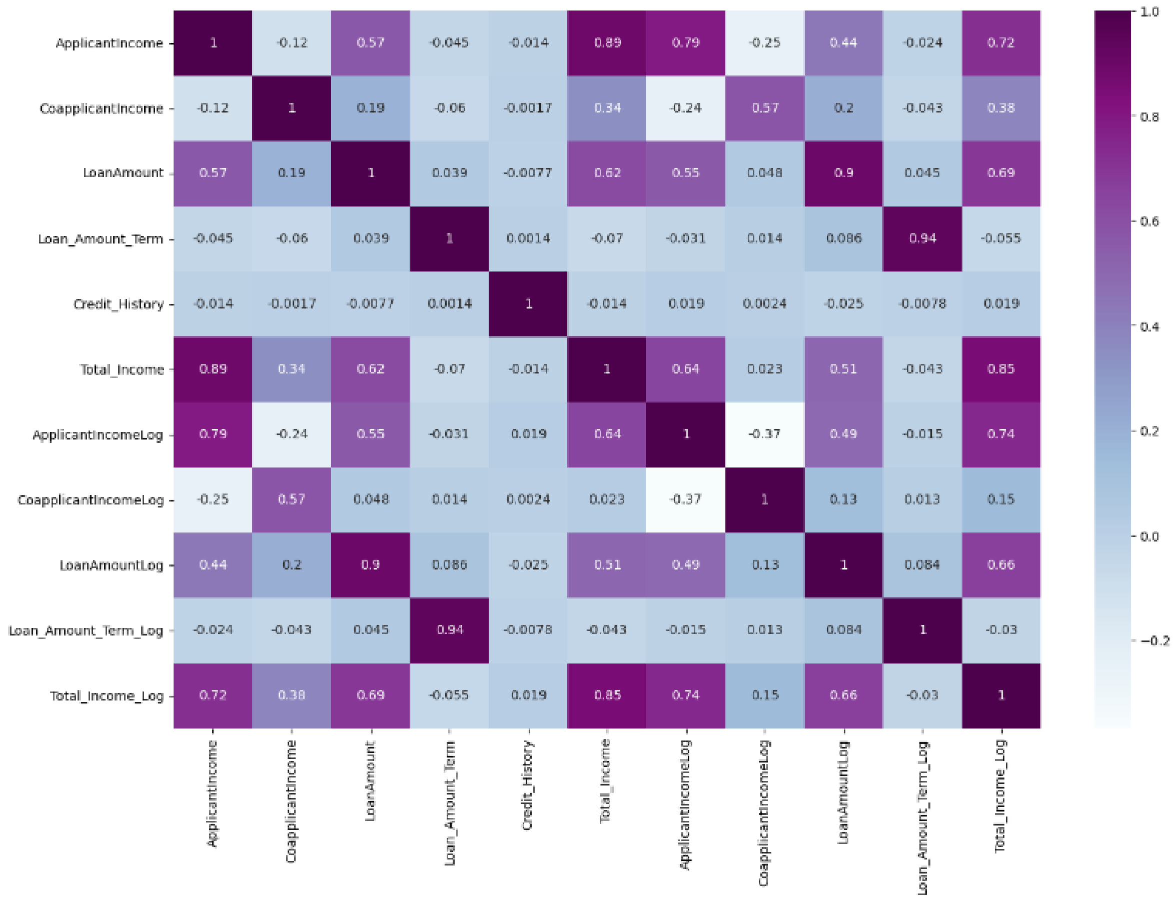
The plot shows that the majority of applicants (over 400) are eligible for a loan. This indicates that the majority of applicants have a good credit score and other financial qualifications that make them eligible for a loan. This could be due to a variety of factors, such as a good credit history, a steady income, and a low debt-to-income ratio.

Correlation:



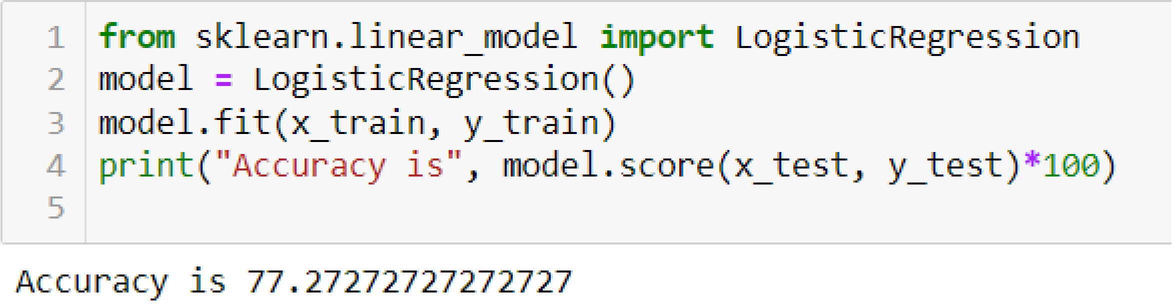
```
corr = df.corr()
plt.figure(figsize=(15,10))
sns.heatmap(corr, annot = True, cmap="BuPu")
```

```
<AxesSubplot:>
```



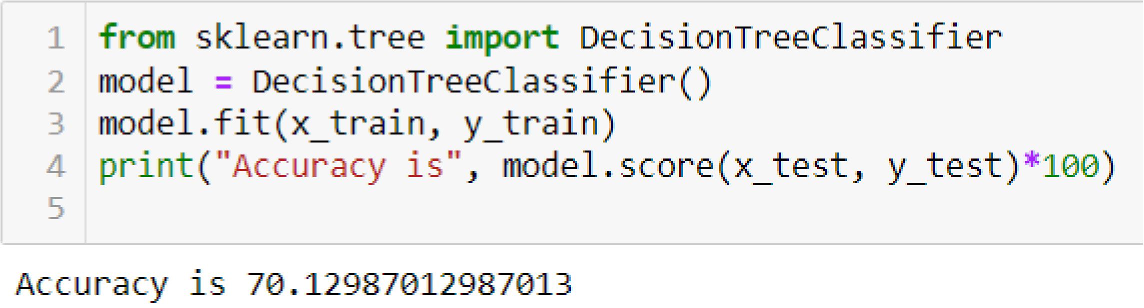
Machine Learning(Models):

Five models were created-Logistic Regression, Decision Tree, Naïve Bayes, SVM, Random Forest. Out of these models created, Random Forest gave the best accuracy.



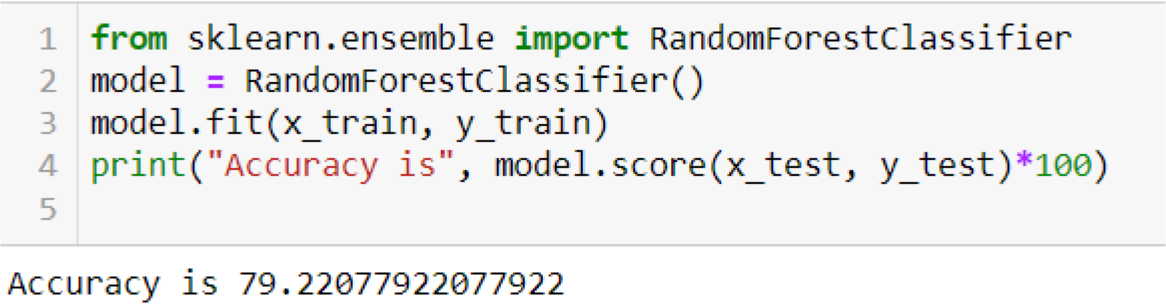
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train, y_train)
print("Accuracy is", model.score(x_test, y_test)*100)

```

```
Accuracy is 77.27272727272727
```



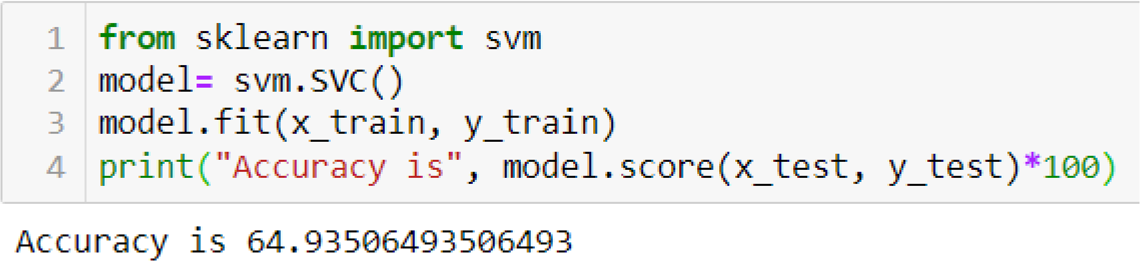
```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(x_train, y_train)
print("Accuracy is", model.score(x_test, y_test)*100)

```

```
Accuracy is 70.12987012987013
```



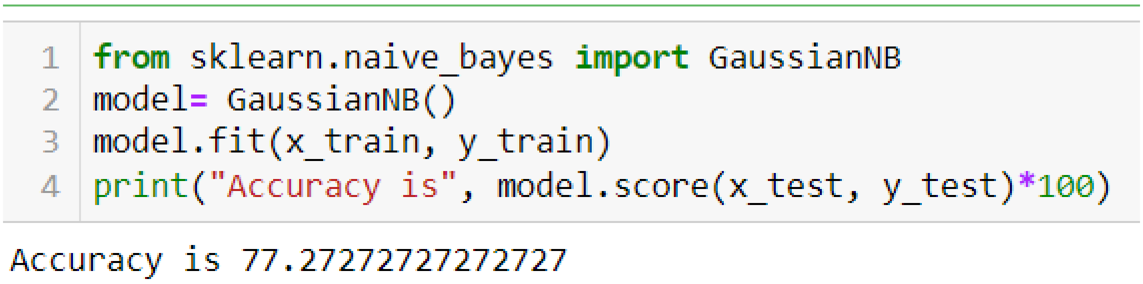
```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(x_train, y_train)
print("Accuracy is", model.score(x_test, y_test)*100)

```

```
Accuracy is 79.22077922077922
```



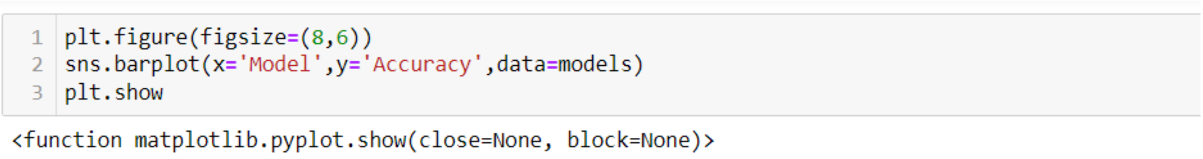
```
from sklearn import svm
model= svm.SVC()
model.fit(x_train, y_train)
print("Accuracy is", model.score(x_test, y_test)*100)
```

```
Accuracy is 64.93506493506493
```



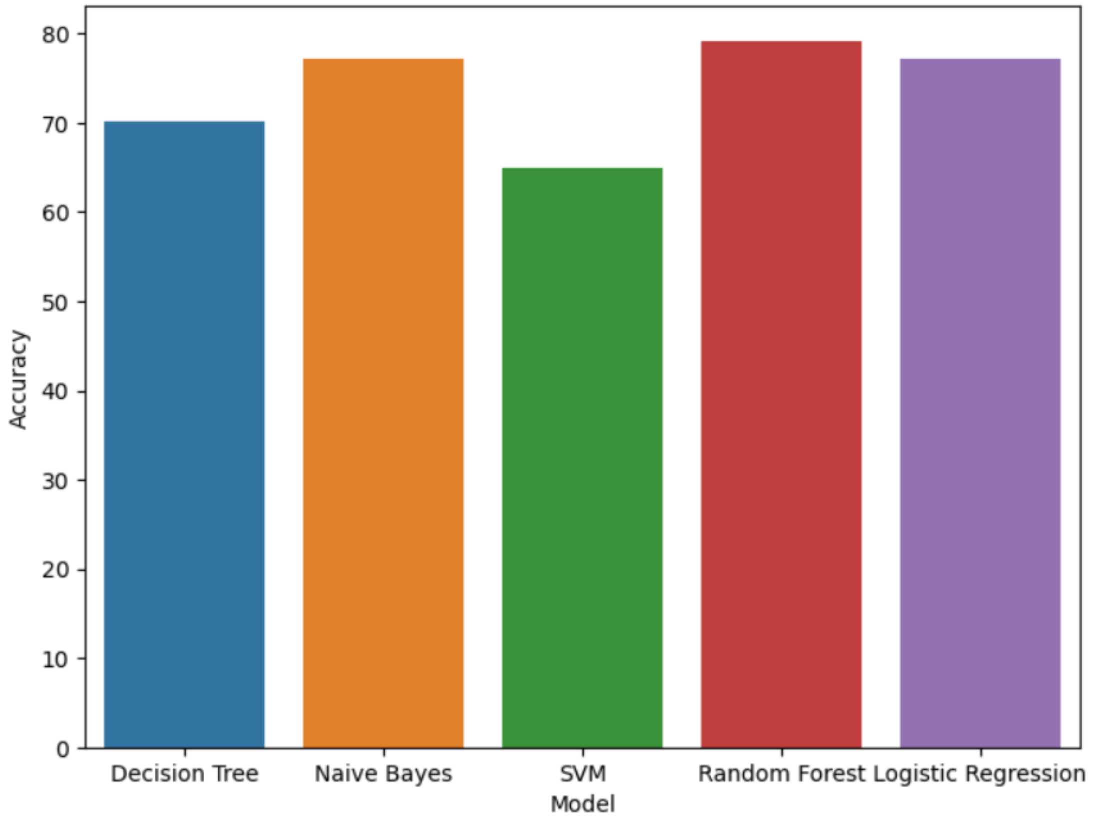
```
from sklearn.naive_bayes import GaussianNB
model= GaussianNB()
model.fit(x_train, y_train)
print("Accuracy is", model.score(x_test, y_test)*100)
```

```
Accuracy is 77.27272727272727
```



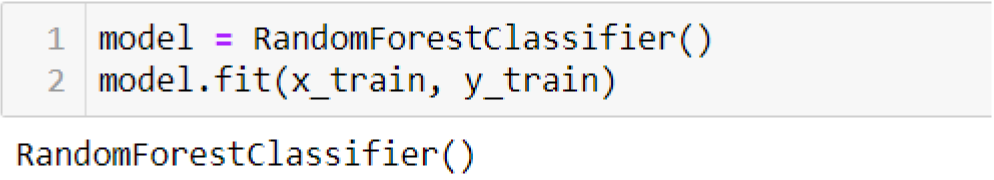
```
plt.figure(figsize=(8,6))
sns.barplot(x='Model',y='Accuracy',data=models)
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



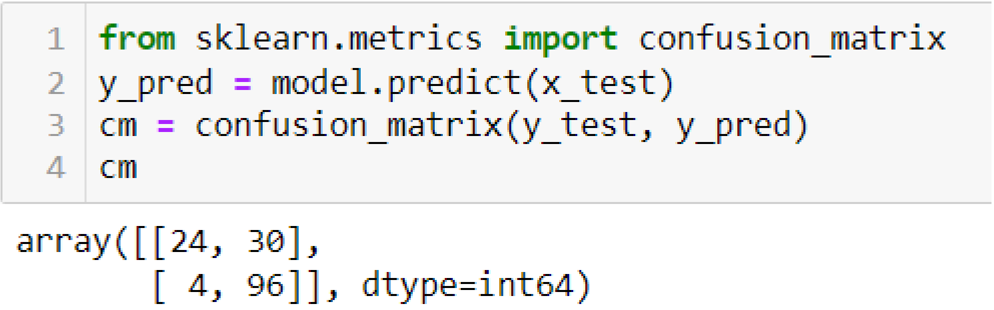
Confusion Matrix:

Since the Random Forest model provided the best accuracy, therefore testing was done on it for the appropriate predictions.



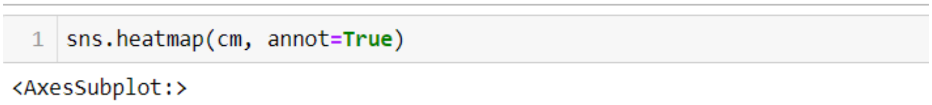
```
model = RandomForestClassifier()
model.fit(x_train, y_train)
```

```
RandomForestClassifier()
```



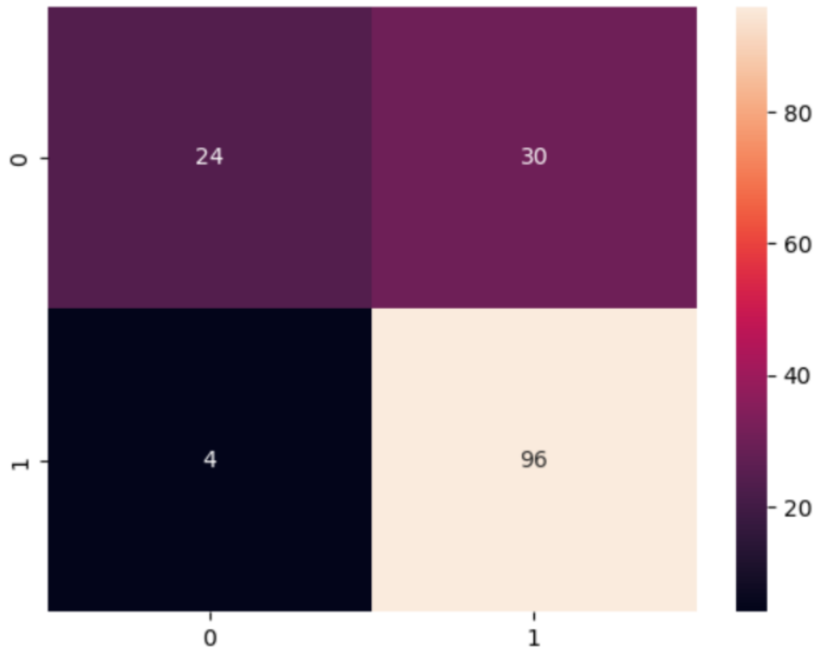
```
from sklearn.metrics import confusion_matrix
y_pred = model.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
cm
```

```
array([[24, 30],
       [ 4, 96]], dtype=int64)
```



```
sns.heatmap(cm, annot=True)
```

```
<AxesSubplot:>
```



Conclusion:

Based on the analysis of the loan prediction dataset, it was found that the Random Forest algorithm gave the best accuracy for predicting loan approval. This indicates that it is a suitable model for financial institutions to use in order to minimize the risk of approving loans to individuals who are unlikely to pay them back.

In conclusion, loan prediction analysis is an important task for financial institutions to minimize the risk of loan default and make informed decisions about loan approvals. The most significant factors for loan prediction analysis typically include credit score, income, debt-to-income ratio, employment history, loan amount, loan term, and purpose of the loan.

By analyzing these factors, financial institutions can better assess the creditworthiness of loan applicants and reduce the risk of loan defaults. However, the relative importance of these factors can vary depending on the specific dataset and modeling techniques used. Therefore, it is important for financial institutions to carefully consider the factors that are most relevant to their specific loan prediction analysis and continually refine their models as new data becomes available.

To improve the loan approval process and reduce the risk for financial institutions, the following recommendations can be considered:

Collect more relevant data: In order to improve the accuracy of the loan prediction model, it is important to collect more relevant data that can be used to assess the creditworthiness of loan applicants. This can include data such as employment history, income, and credit history.

Use alternative data sources: In addition to traditional credit data, financial institutions can also use alternative data sources such as social media data, mobile phone data, and utility data to assess the creditworthiness of loan applicants.

Implement stricter credit policies: Financial institutions can also reduce the risk of loan defaults by implementing stricter credit policies. This can include setting minimum credit score requirements, requiring collateral, and limiting the amount of credit extended to individual borrowers.

Provide financial education: To minimize the risk of loan defaults, financial institutions can also provide financial education to their clients to help them manage their finances and make informed decisions about borrowing.

By implementing these recommendations, financial institutions can improve their loan approval process and reduce the risk of loan defaults, thereby protecting their financial health and stability.