**Loan Approval Prediction**

The aim of this project is to predict whether the loan would be approved by the bank, by analyzing the applicant's information which includes loan amount, tenure, cibil score, education, assests and many other variables. Thorugh this project, we can analyze the factors that affect the loan approval and also predict the loan approval status for a new applicant. Moreover, this will help in providing priority services to the customers who are more likely to get their loan approved.

# About the dataset

The loan approval dataset is a collection of financial records and associated information used to determine the eligibility of individuals or organizations for obtaining loans from a lending institution. It includes various factors such as cibil score, income, employment status, loan term, loan amount, assets value, and loan status. This dataset is commonly used in machine learning and data analysis to develop models and algorithms that predict the likelihood of loan approval based on the given features.

# Data Dictionary

**Variable Description**

loan\_id Unique loan ID

|  |  |
| --- | --- |
| no\_of\_dependents | Number of dependents of the applicant |
| education | Education level of the applicant |
| self\_employed | If the applicant is self-employed or not |
| income\_annum | Annual income of the applicant |
| loan\_amount | Loan amount requested by the applicant |
| loan\_tenure | Tenure of the loan requested by the applicant (in Years) |
| cibil\_score | CIBIL score of the applicant |
| residential\_asset\_value | Value of the residential asset of the applicant |
| commercial\_asset\_value | Value of the commercial asset of the applicant |
| luxury\_asset\_value | Value of the luxury asset of the applicant |
| bank\_assets\_value | Value of the bank asset of the applicant |
| loan\_status | Status of the loan (Approved/Rejected) |

|  |
| --- |
| *# Importing the libraries* **import** numpy **as** np **import** matplotlib.pyplot **as** plt **import** pandas **as** pd **import** seaborn **as** sns |

In [ ]:

|  |  |  |
| --- | --- | --- |
| *# Loading the dataset*  df **=** pd**.**read\_csv('loan\_approval\_dataset.csv') df**.**head() |  |  |
| **loan\_id no\_of\_dependents education self\_employed** | **income\_annum** | **loan\_amount** |

In [ ]:

Out[ ]:

**0** 1 2 Graduate No 9600000 29900000

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | 2 | 0 | Not Graduate | Yes |  | 4100000 | 12200000 |
| **2** | 3 | 3 | Graduate | No |  | 9100000 | 29700000 |
| **3** | 4 | 3 | Graduate | No |  | 8200000 | 30700000 |
| **4** | 5 | 5 | Not Graduate | Yes |  | 9800000 | 24200000 |

# Data Preprocessing

In [ ]: *# Checking the shape of the dataset* df**.**shape

Out[ ]: (4269, 13)

Removing the unnecessary load\_id as it is an identifier column

In [ ]: df**.**drop(columns**=**'loan\_id', inplace**=True**)

In [ ]: *# Checking for null/missing values* df**.**isnull()**.**sum()

Out[ ]: 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 [ ]: *# Checking the data types of the columns* df**.**dtypes

Out[ ]: 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

The dataset has 4 kinds of assests that are - Residential, Commericial, Luxury and Bank. I am categorizing these assets in to two category i.e. Movable and Immovable assets. The Residential and Commericial assest would be added to the Immovable assets and Luxury and Bank assets would be added to the Movable assets.

In [ ]: *# Movable Assets* df['Movable\_assets'] **=** df[' bank\_asset\_value'] **+** df[' luxury\_assets\_value']

*#Immovable Assets* df['Immovable\_assets'] **=** df[' residential\_assets\_value'] **+** df[' commercial\_asset

|  |  |
| --- | --- |
| *# Drop columns*  df**.**drop(columns**=**[' bank\_asset\_value',' luxury\_assets\_value', | ' residential\_asset |
| Descriptive Statistics |  |
| df**.**describe() |  |
| **no\_of\_dependents income\_annum loan\_amount loan\_term** | **cibil\_score Mov** |

In [ ]:

In [ ]:

Out[ ]:

**count** 4269.000000 4.269000e+03 4.269000e+03 4269.000000 4269.000000 4.2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **mean** |  | 2.498712 | 5.059124e+06 | 1.513345e+07 | 10.900445 | 599.936051 |  |
| **std** |  | 1.695910 | 2.806840e+06 | 9.043363e+06 | 5.709187 | 172.430401 |  |
| **min** |  | 0.000000 | 2.000000e+05 | 3.000000e+05 | 2.000000 | 300.000000 |  |
| **25%** |  | 1.000000 | 2.700000e+06 | 7.700000e+06 | 6.000000 | 453.000000 |  |
| **50%** |  | 3.000000 | 5.100000e+06 | 1.450000e+07 | 10.000000 | 600.000000 |  |
| **75%** |  | 4.000000 | 7.500000e+06 | 2.150000e+07 | 16.000000 | 748.000000 |  |
| **max** |  | 5.000000 | 9.900000e+06 | 3.950000e+07 | 20.000000 | 900.000000 |  |

2.0

1.1

3.0

1.0

1.9

2.9

5.3

|  |
| --- |
| df**.**head() |

In [ ]:

**no\_of\_dependents**

**education**

**self\_employed**

**income\_annum**

**loan\_amount**

**loan\_term**

**0**

2

Graduate

No

9600000

29900000

1

**1**

0

Not

Graduate

Yes

4100000

12200000

**2**

3

Graduate

No

9100000

29700000

2

**3**

3

Graduate

No

8200000

30700000

**4**

5

Not

Graduate

Yes

9800000

24200000

2

Out[ ]:

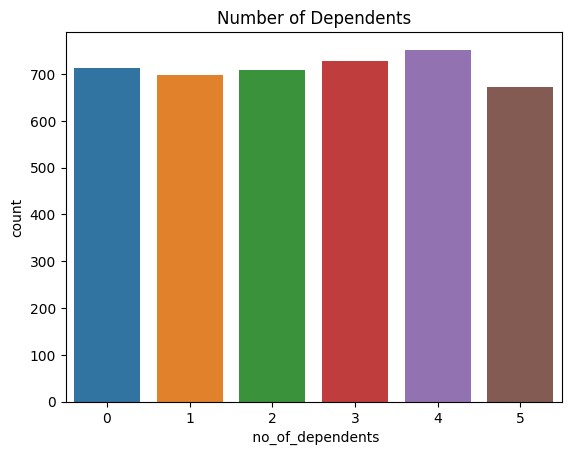
# Exploratory Data Analysis

In the exploratory data analysis, I will be looking at the distribution of the data across the varaiables, followed by relationship between the independent and target variable and the correlation among the variables. Through the visualization, I will be able to understand the possible trends and patterns in the data and come to know about the hidden insights of the data.

## Number of Dependents

In [ ]: sns**.**countplot(x **=** ' no\_of\_dependents', data **=** df)**.**set\_title('Number of Dependent

Out[ ]: Text(0.5, 1.0, 'Number of Dependents')

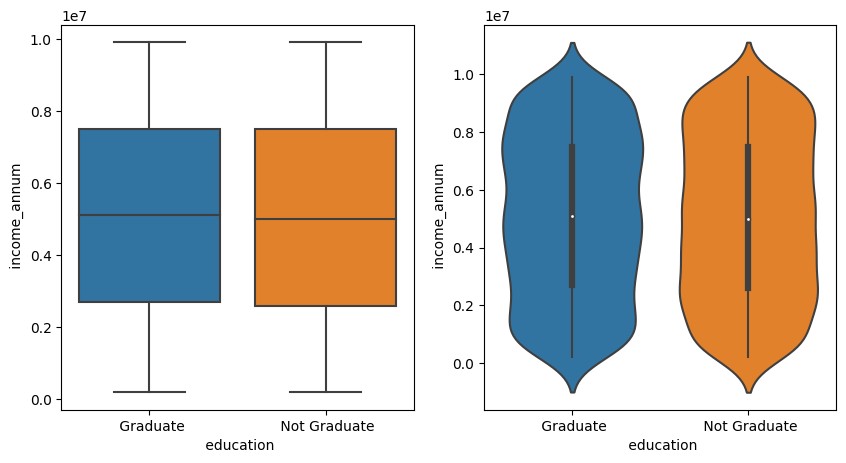


This graph shows the number of dependent indivduals on the loan applicant. There is not much difference in the number of dependents, however, there are more applicants with 4 and 3 dependents than the other categories. Since the number of dependents increases the disposable income of the applicant decreases. So I assume that that the number of applicants with 0 or 1 dependent will have higher chances of loan approval.

## Education and Income

In [ ]: fig, ax **=** plt**.**subplots(1,2,figsize**=**(10, 5)) sns**.**boxplot(x **=** ' education', y **=** ' income\_annum', data **=** df, ax**=**ax[0]) sns**.**violinplot(x **=** ' education', y **=** ' income\_annum', data **=** df, ax**=**ax[1])

Out[ ]: <Axes: xlabel=' education', ylabel=' income\_annum'>

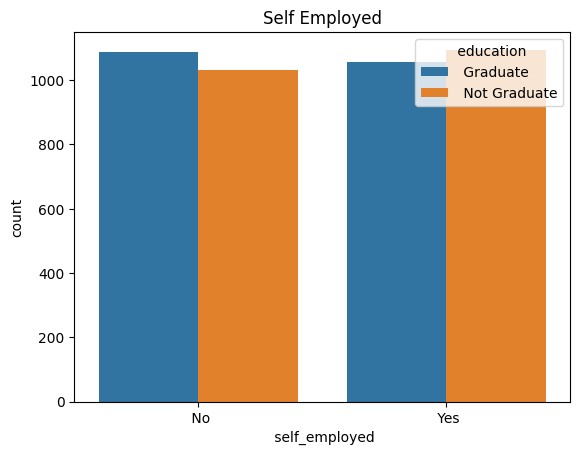


These two graphs - boxplot and violinplot visualizes the education of applicants along with their annual income. The boxplot shows some interesting fact that both the graduates and non-graduates have nearly same median income with very small increase in income of graduates. Moreover the violinplot shows the distribution of income among the graduates and non graduate applicants, where we can see that non graduate applicants have a even distribution between income 2000000 and 8000000, whereas there is a uneven distribution among the graduates with more applicants having income between 6000000 and 8000000. Since there is not much change in annual income of graduates and non graduates, I assume that education does not play a major role in the approval of loan.

## Employment Status and Education

In [ ]: sns**.**countplot(x**=**' self\_employed', data **=** df, hue **=** ' education')**.**set\_title('Self

Out[ ]: Text(0.5, 1.0, 'Self Employed')

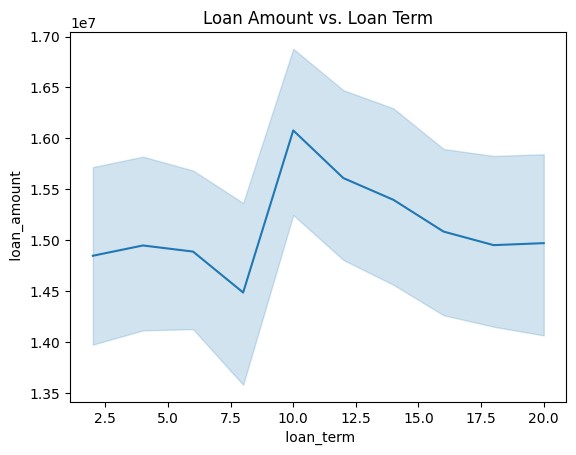


This graph shows the number of self employed applicants along with their education. From the educational prepespective the majority of the graducate applicants are not self employed wheareas majority of the non-graduates are self employed. This means that graduates applicants are more likely to be salaried employees and non-graduates are more likely to be self employed. This could be a determining factor in loan approval because salaried employees are more likely to have a stable income and hence are more likely to pay back the loan as compared to self employed applicants whose income may not be stable. But this could also be possible that the self employed applicants are earning more than the salaried employees and hence are more likely to pay back the loan. This is a very important factor to consider while predicting the loan approval.

## Loan Amount and Tenure

In [ ]: sns**.**lineplot(x **=** ' loan\_term', y **=** ' loan\_amount', data **=** df)**.**set\_title('Loan Am

Out[ ]: Text(0.5, 1.0, 'Loan Amount vs. Loan Term')



This line plot shows the trend between the loan amount and the loan tenure. Between the loan tenure of 2.5 - 7.5 years the loan amount is between 1400000 - 15500000.

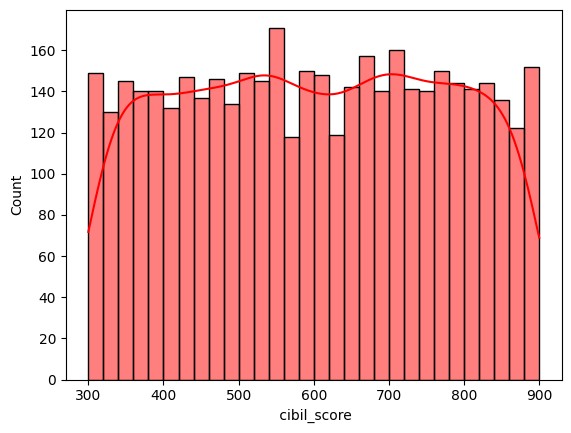
However the loan amount is significantly higher for the loan tenure of 10 years.

## CIBIL Score Distribution

|  |
| --- |
| sns**.**histplot(df[' cibil\_score'], bins **=** 30, kde **=** **True**, color **=** 'red') |

In [ ]:

Out[ ]: <Axes: xlabel=' cibil\_score', ylabel='Count'>



Before looking at the cibil score, lets have a look at the cibil score ranges and their meaning.

**Cibil Score Meaning**

300-549 Poor

|  |  |
| --- | --- |
| 550-649 | Fair |
| 650-749 | Good |
| 750-799 | Very Good |
| 800-900 | Excellent |

Source: [godigit.com](https://www.godigit.com/finance/credit-score/ranges-of-credit-score)

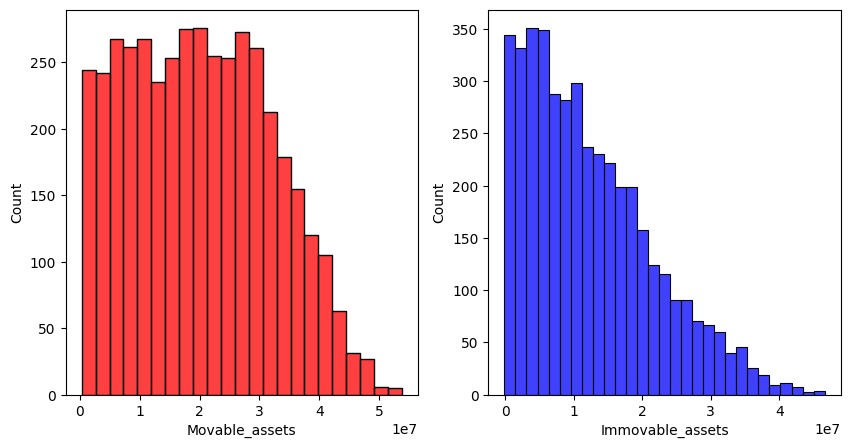
Taking the above table as a reference for the cibil score quality, majority of the customers have cibil score below 649, which affects their loan application. However there are many applicants with cibil score above 649, which is a good sign for the bank. The bank can target these customers and provide them with priority services. The bank can also provide them with special offers and discounts to attract them to take loans from the bank. From this, I build a hypothesis that the customers with cibil score above 649 are more likely to get their loan approved.

## Asset Distribution

|  |
| --- |
| fig, ax **=** plt**.**subplots(1,2,figsize**=**(10,5)) sns**.**histplot(df['Movable\_assets'], ax**=**ax[0], color**=**'red') sns**.**histplot(df['Immovable\_assets'], ax**=**ax[1], color**=**'blue') |

In [ ]:

Out[ ]: <Axes: xlabel='Immovable\_assets', ylabel='Count'>



Assets play a major role in loan application. They provides a security to the bank that the person will repay the loan. Looking at the assets, as eralier mentionedI have categorized them in movable and immovable assets. The above graphs shows the distribution of movable and immovable assets in the dataset.

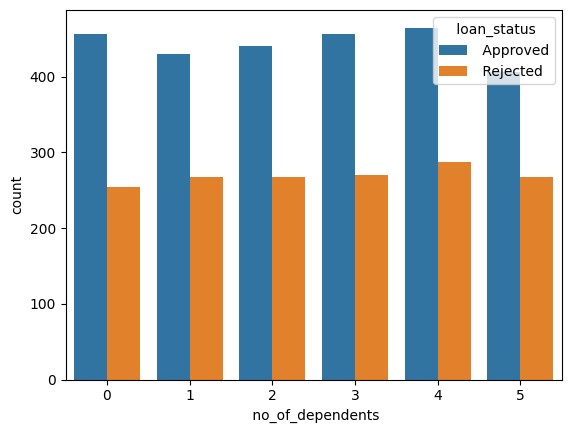
Looking at the movable assets which include bank assets and luxury assets, majority of the applicants have less than 30 million and there is a slight trend of decreasing number of applicants as the movable assets increases. Coming to the immovable assets, which include residential assets and commercial assets, majority of the applicants have less than 15 million of immovable assets and there is a strong trend of decreasing number of applicants as the immovable assets increases after 20 million.

Till now in the EDA, I have explored the distribution of data across the various features as well as relationship between the some of the variables as well and made some assumptions and hypothesis. Now, in order to prove my assumptions and hypothesis I will be looking at the visualization of the relation between the independent variables and the target variable.

## Number of Dependants Vs Loan Status

In [ ]: sns**.**countplot(x **=** ' no\_of\_dependents', data **=** df, hue **=** ' loan\_status')

Out[ ]: <Axes: xlabel=' no\_of\_dependents', ylabel='count'>

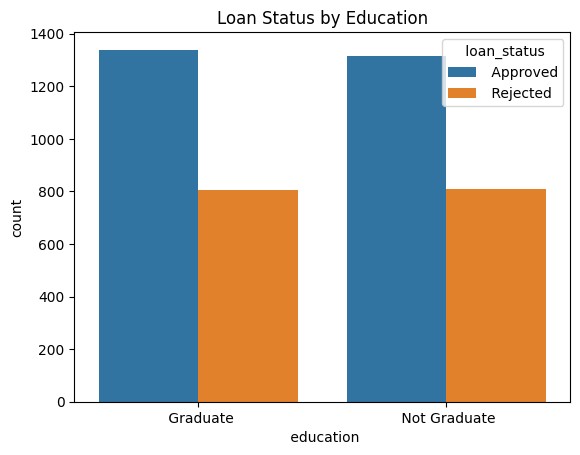


My hypothesis regarding the loan approval based on number of dependents has mixed results. First the hypothesis was somewhat true regarding the rejection chances, the number of loan rejection increases with increase in number of dependents. But the hypothesis was not true regarding the approval chances, the number of loan approval decreases with increase in number of dependents as per my hypothesis. But according to this graph, there has been no major change in the loan approval count with increase in number of dependents. So, my hypothesis regarding the loan approval based on number of dependents is not true.

## Education Vs Loan Status

In [ ]: sns**.**countplot(x **=** ' education', hue **=** ' loan\_status', data **=** df)**.**set\_title('Loan

Out[ ]: Text(0.5, 1.0, 'Loan Status by Education')



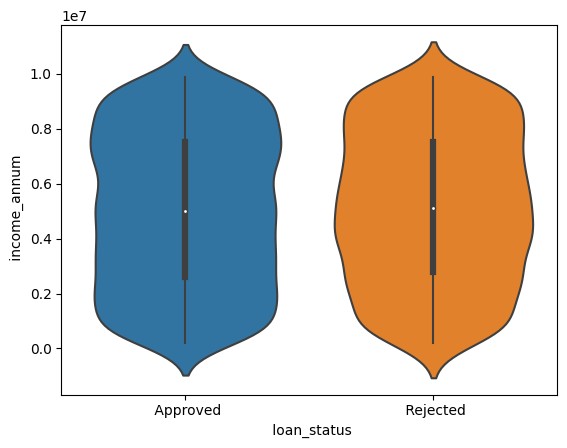
My hypothesis regarding the education not being factor in loan approval was right. The graph shows very minor difference between loan approval and rejection count for the graduate and non graduate applicants. The difference is not significant enough.

## Annual Income vs Loan Status

|  |
| --- |
| sns**.**violinplot(x**=**' loan\_status', y**=**' income\_annum', data**=**df) |

In [ ]:

Out[ ]: <Axes: xlabel=' loan\_status', ylabel=' income\_annum'>

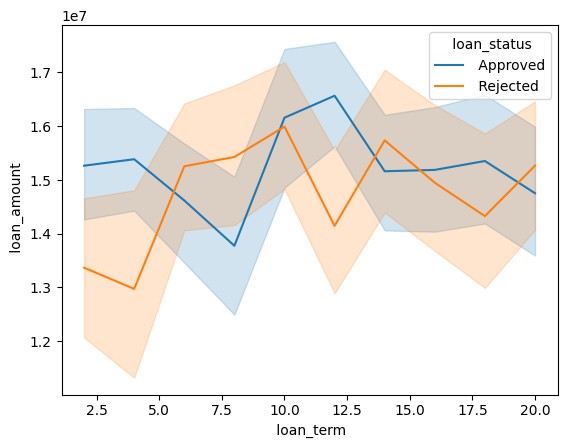


On the whole, there has been no major difference between the annual incomes of the applicant with approved or rejected loan. But still, the approved loan applicants tend to have a higher annual income than the rejected loan applicants which is visible from the violin plot where the approved loan applicants have a higher density in the annual income near 8 miilion annual income.

## Loan amount & tenure Vs Loan Status

In [ ]: sns**.**lineplot(x**=**' loan\_term', y**=**' loan\_amount', data**=**df, hue**=**' loan\_status')

Out[ ]: <Axes: xlabel=' loan\_term', ylabel=' loan\_amount'>



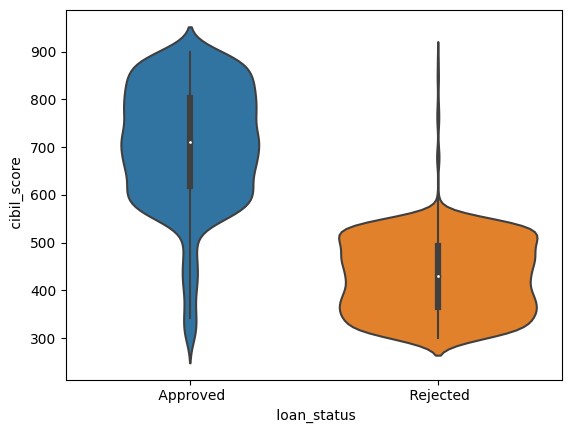
This graph shows the relation between loan amount, loan tenure and loan status.

Generally, the approved loans tend have higher amount and shorter repayment tenure. The rejected loans tend to have lower amount and longer repayment tenure. This could be a result of the bank's policy to reject loans with longer repayment tenure. The bank may also reject loans with lower amount as they may not be profitable for the bank.

## CIBIL Score Vs Loan Status

In [ ]: sns**.**violinplot(x**=**' loan\_status', y**=**' cibil\_score', data**=**df)

Out[ ]: <Axes: xlabel=' loan\_status', ylabel=' cibil\_score'>

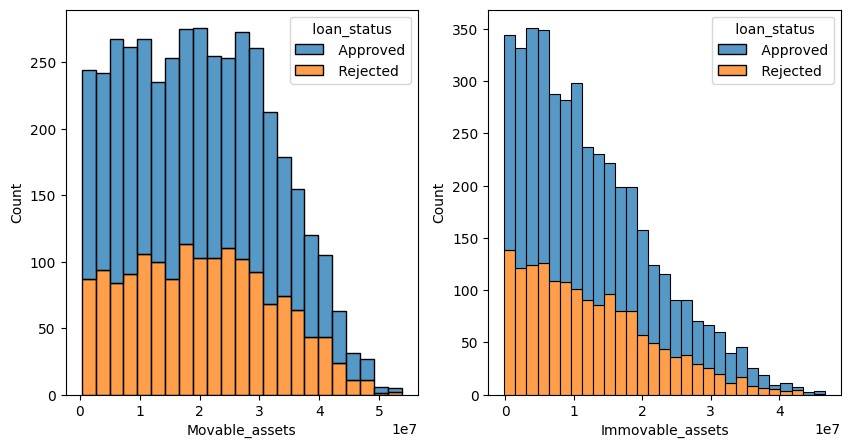


My hypothesis regarding the cibil score and loan approval is absolutely correct. It is evident through the violinplot, where the there is a high distribution above 600 cibil score from the loan approved category. The distribution of the loan not approved category is more spread out and has cibil score less than 550. This also proves my assumption that majority of the applicants have a poor/fair cibil score which affects their loan approval. Hence, having a high cibil score particularly grater than 600 would definitely increase the chances of loan approval.

## Assets Vs Loan Status

In [ ]: fig, ax **=** plt**.**subplots(1,2,figsize**=**(10,5)) sns**.**histplot(x **=** 'Movable\_assets', data **=** df, ax**=**ax[0], hue **=** ' loan\_status', m sns**.**histplot(x **=** 'Immovable\_assets', data **=** df, ax**=**ax[1], hue **=** ' loan\_status'

Out[ ]: <Axes: xlabel='Immovable\_assets', ylabel='Count'>



Assets provide security to the bank against which the loan is issued. These two graph visualizes the relation between the movable and immovable assets along with the loan status. The both graph shows that, with increase in the assets the chances of loan approval increases and rejection decreases. The graph also shows that, the movable assets are more than the immovable assets.

# Data Preprocessing 2

## Label Encoding the categorical variables

In [ ]: *# Label Encoding* df[' education'] **=** df[' education']**.**map({' Not Graduate':0, ' Graduate':1}) df[' self\_employed'] **=** df[' self\_employed']**.**map({' No':0, ' Yes':1}) df[' loan\_status'] **=** df[' loan\_status']**.**map({' Rejected':0, ' Approved':1})

In [ ]: df**.**head()

Out[ ]: **no\_of\_dependents education self\_employed income\_annum loan\_amount loan\_term**

**0** 2 1 0 9600000 29900000 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | 0 | 0 | 1 | 4100000 | 12200000 |  |
| **2** | 3 | 1 | 0 | 9100000 | 29700000 |  |
| **3** | 3 | 1 | 0 | 8200000 | 30700000 |  |
| **4** | 5 | 0 | 1 | 9800000 | 24200000 |  |

2

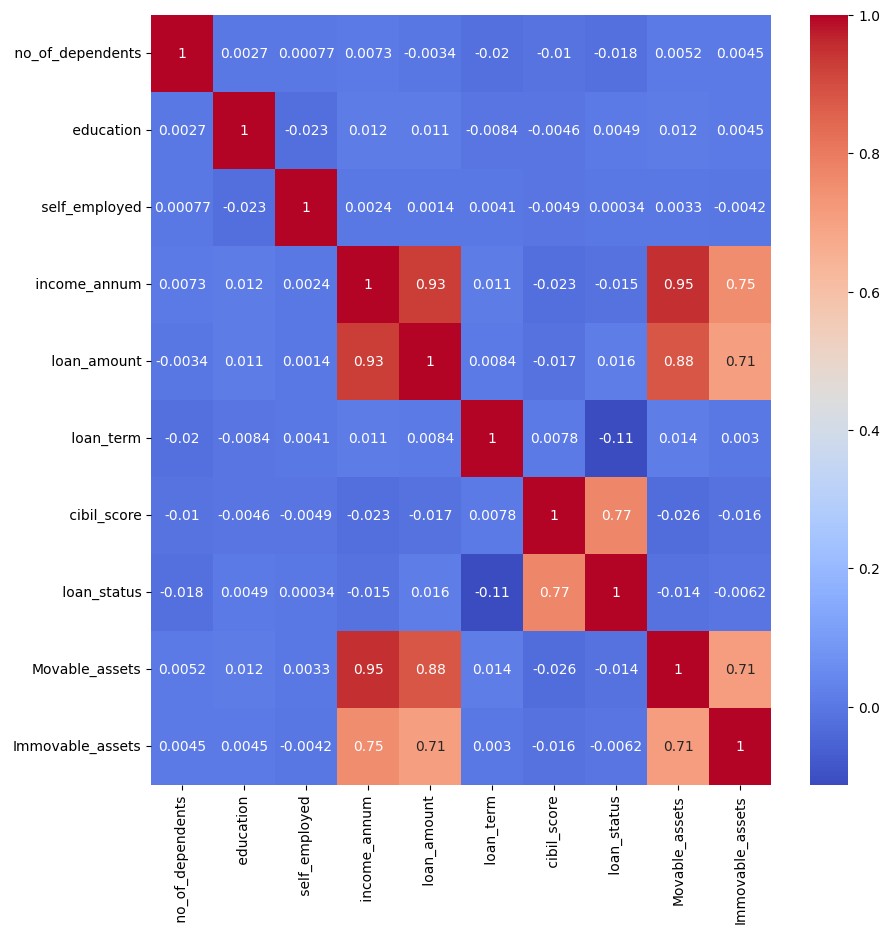
2

# Coorelation Matrix Heatmap

|  |
| --- |
| plt**.**figure(figsize**=**(10,10))  sns**.**heatmap(df**.**corr(),annot **=** **True**,cmap**=**'coolwarm') |

In [ ]:

Out[ ]: <Axes: >



This coorelation matrix heatmap has the folowing strong correlations:

1. Movable Assets and Immovable Assets
2. Income and Movable Assets
3. Income and Immovable Assets
4. Movable Assets and Loan Amount
5. Immovable Assets and Loan Amount
6. Loan Status and Cibil Score
7. Loan Amount and Income

The coorelation between the movable and immovable assets is justified because both come under assets and its obvious that person with more movable assets will have more immovable assets and vice versa. Same is with Income and Movables and Immovale assets. The person with greater income will have greater assets.

Now, I will be exploring the coorleation between Assets and Loan Amount, and also between Income and Loan Amount. The relation between the loan status and cibil score is already explored in the previous section.

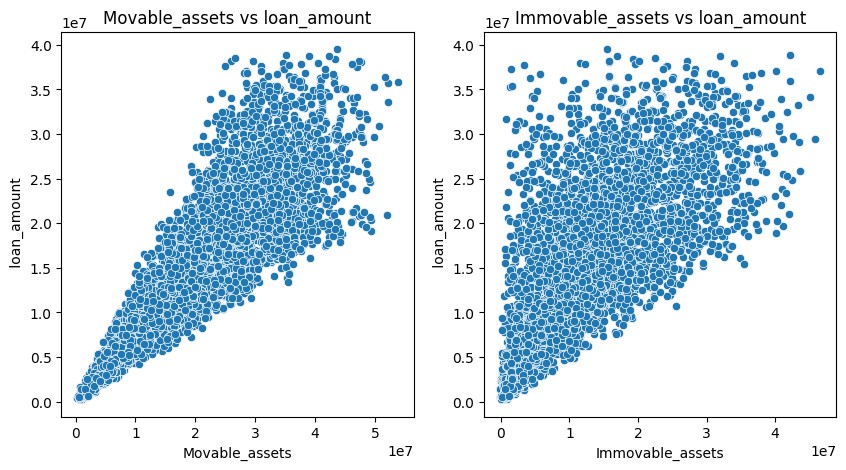
## Assets Vs Loan Amount

In [ ]:

|  |
| --- |
| fig, ax **=** plt**.**subplots(1,2,figsize**=**(10, 5)) sns**.**scatterplot(x**=**'Movable\_assets', y **=** ' loan\_amount', data **=** df, ax**=**ax[0])**.**set sns**.**scatterplot(x**=**'Immovable\_assets', y **=** ' loan\_amount', data **=** df, ax**=**ax[1])**.** |

s

Out[ ]: Text(0.5, 1.0, 'Immovable\_assets vs loan\_amount')



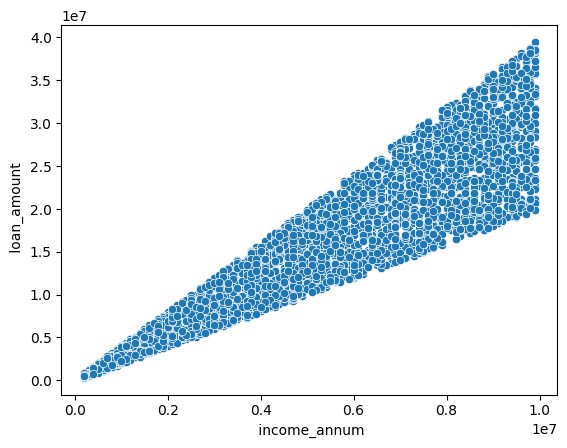
The loan amount has positive relation with movable and immovable assets. The more the assets, the more the loan amount issued by the bank.

## Loan Amount Vs Income

|  |
| --- |
| sns**.**scatterplot(x**=**' income\_annum', y **=** ' loan\_amount', data **=** df) |

In [ ]:

Out[ ]: <Axes: xlabel=' income\_annum', ylabel=' loan\_amount'>



The loan amount and applicant's annual income have a very direct relation between them. The higher the income, the higher the loan amount. This is because the applicant's income is the main factor in deciding the how much loan needed.

# Train Test Split

|  |
| --- |
| **from** sklearn.model\_selection **import** train\_test\_split  X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(df**.**drop(' loan\_status', axis |

In [ ]:

# Model Building

I will be using the following machine learning models to predcit the loan approval status:

1. Decision Tree Classifier
2. Random Forest Classifier

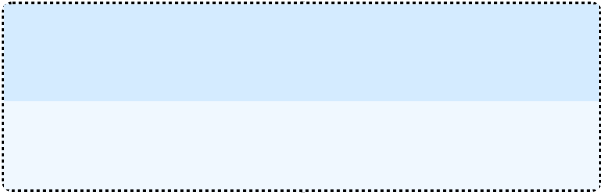
## Decision Tree Classifier

|  |
| --- |
| **from** sklearn.tree **import** DecisionTreeClassifier  *# Create decision tree object* dtree **=** DecisionTreeClassifier() |

In [ ]:

|  |
| --- |
| *# Trainign the model using the training data* dtree**.**fit(X\_train, y\_train) |

In [ ]:

Out[ ]: ▾DecisionTreeClassifier

DecisionTreeClassifier()

In [ ]: *# Training Accuracy* dtree**.**score(X\_train, y\_train)

|  |  |
| --- | --- |
| Out[ ]: | 1.0 |

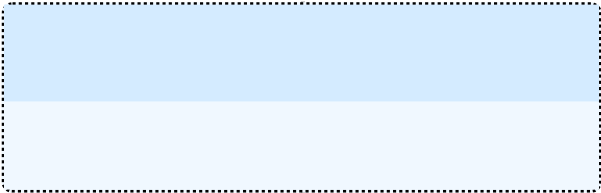
In [ ]: *# Predicting the Loan Approval Status* dtree\_pred **=** dtree**.**predict(X\_test)

## Random Forest Classifier

|  |
| --- |
| **from** sklearn.ensemble **import** RandomForestClassifier  *# Create a random forest classifier* rfc **=** RandomForestClassifier() |

In [ ]:

In [ ]: *# Training the model using the training data* rfc**.**fit(X\_train, y\_train)

Out[ ]: ▾RandomForestClassifier

RandomForestClassifier()

In [ ]: *# Training Accuracy* rfc**.**score(X\_train, y\_train)

|  |  |
| --- | --- |
| Out[ ]: | 1.0 |

In [ ]: *# Predicting the Loan Approval Status*

rfc\_pred **=** rfc**.**predict(X\_test)

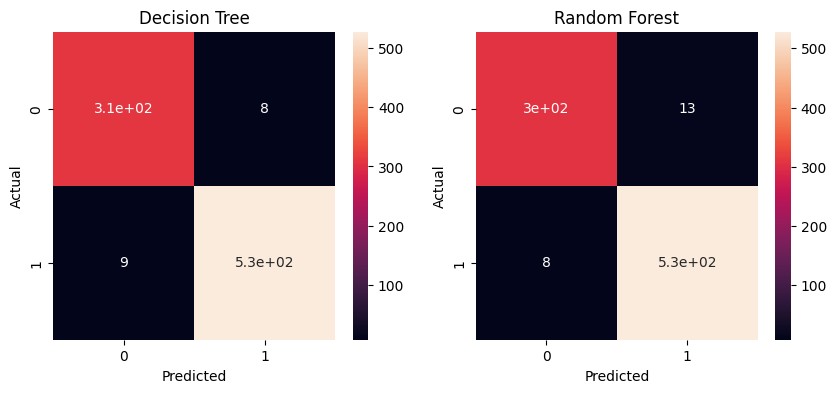
# Model Evalution

## Confusion Matrix

In [ ]: **from** sklearn.metrics **import** confusion\_matrix

fig, ax **=** plt**.**subplots(1,2,figsize**=**(10,4)) sns**.**heatmap(confusion\_matrix(y\_test, dtree\_pred), annot**=True**, ax**=**ax[0])**.**set\_titl ax[0]**.**set\_xlabel('Predicted') ax[0]**.**set\_ylabel('Actual') sns**.**heatmap(confusion\_matrix(y\_test, rfc\_pred), annot**=True**, ax**=**ax[1])**.**set\_title( ax[1]**.**set\_xlabel('Predicted') ax[1]**.**set\_ylabel('Actual')

Out[ ]: Text(518.4494949494949, 0.5, 'Actual')



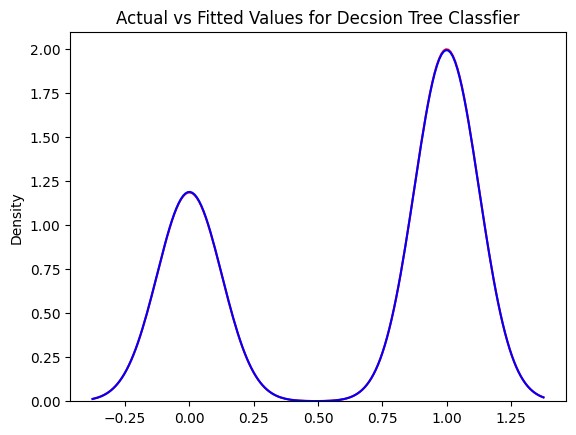
The above confusion matrix heatmap visualizes the the true positive and true negative value counts in both the machine learning models. The decision tree classfier has only 17 false positve and negative valyes where has random forest classifier has 21 false postive and negative values. The decision tree classifier has a better accuracy compared to random forest classifier.

## Distribution Plot

|  |
| --- |
| ax **=** sns**.**distplot( x **=** y\_test, hist **=** **False**, color **=** "r", label **=** "Actual Value" sns**.**distplot( x **=** dtree\_pred, hist **=** **False**, color **=** "b", label **=** "Fitted Values" plt**.**title('Actual vs Fitted Values for Decsion Tree Classfier') |

In [ ]:

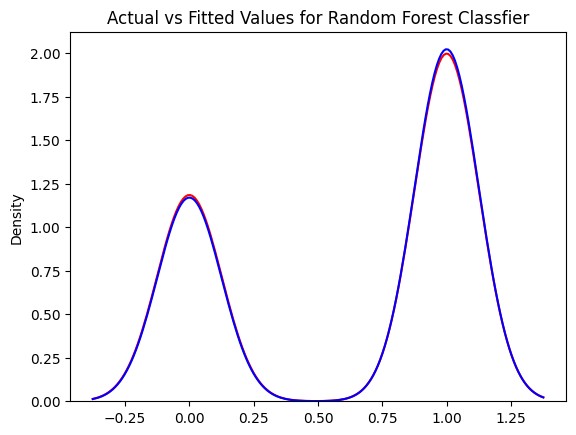
Out[ ]: Text(0.5, 1.0, 'Actual vs Fitted Values for Decsion Tree Classfier')



|  |
| --- |
| ax **=** sns**.**distplot( x **=** y\_test, hist **=** **False**, color **=** "r", label **=** "Actual Value" sns**.**distplot( x **=** rfc\_pred, hist **=** **False**, color **=** "b", label **=** "Fitted Values", plt**.**title('Actual vs Fitted Values for Random Forest Classfier') |

In [ ]:

Out[ ]: Text(0.5, 1.0, 'Actual vs Fitted Values for Random Forest Classfier')



The distribution plot of both the models are almost same. There is very minute difference in the distribution density of the predicted and actual values in the random forest classifer. However, in case of decision tree classifier, the distribution density of the predicted values clearly overlaps with the actual values. Hence, we can say that the decision tree classifier is a better model than the random forest classifier for this dataset. **Classification Report**

|  |
| --- |
| **from** sklearn.metrics **import** classification\_report  print(classification\_report(y\_test, dtree\_pred)) print(classification\_report(y\_test, rfc\_pred)) |

In [ ]:

precision recall f1-score support

0 0.97 0.97 0.97 318 1 0.99 0.98 0.98 536

accuracy 0.98 854 macro avg 0.98 0.98 0.98 854 weighted avg 0.98 0.98 0.98 854 precision recall f1-score support

0 0.97 0.96 0.97 318 1 0.98 0.99 0.98 536

accuracy 0.98 854 macro avg 0.98 0.97 0.97 854 weighted avg 0.98 0.98 0.98 854

|  |
| --- |
| **from** sklearn.metrics **import** r2\_score, mean\_squared\_error, mean\_absolute\_error  *# Decision Tree Classifier*  print('R2 score: ', r2\_score(y\_test, dtree\_pred)) print('Mean Squared Error: ', mean\_squared\_error(y\_test, dtree\_pred)) print('Mean Absolute Error: ', mean\_absolute\_error(y\_test, dtree\_pred)) print('\n')  *# Random Forest Classifier*  print('R2 score: ', r2\_score(y\_test, rfc\_pred)) print('Mean Squared Error: ', mean\_squared\_error(y\_test, rfc\_pred)) print('Mean Absolute Error: ', mean\_absolute\_error(y\_test, rfc\_pred)) |

In [ ]:

R2 score: 0.9148244625926969

Mean Squared Error: 0.01990632318501171

Mean Absolute Error: 0.01990632318501171

R2 score: 0.8947831596733314

Mean Squared Error: 0.02459016393442623

Mean Absolute Error: 0.02459016393442623

From all the above metrics, graphs and reports, I conclude that descision tree classifier is a better machine learning model to predict the loan approval status of a person.

# Conclusion

From the exploratory data analysis, we can conclude that the following factors are important for the approval of loan:

CIBIL Score: People with higher CIBIL score have higher chances of loan approval Number of Dependents: People with more number of dependents have less chances of loan approval

Assets: People with more assets ( including movable and immovable) have higher chances of loan approval

Loan Amount and Tenure: People with higher loan amount and lower tenure have more chances of loan approval

Coming to the machine learning models, I have used Decision Tree Classifier and

Random Forest Classifier. Both the models have given excellent results having accuracies - 91.4 % and 89.4 % repectively. But the decision tree classifier has yielded better results than the random forest classifier.