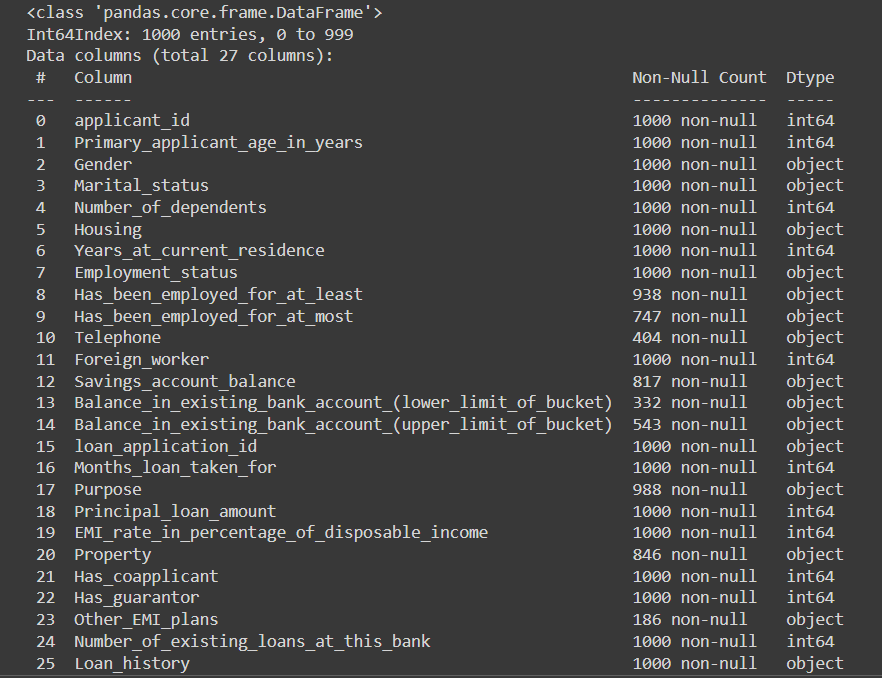
Loan Default Risk Classification Project

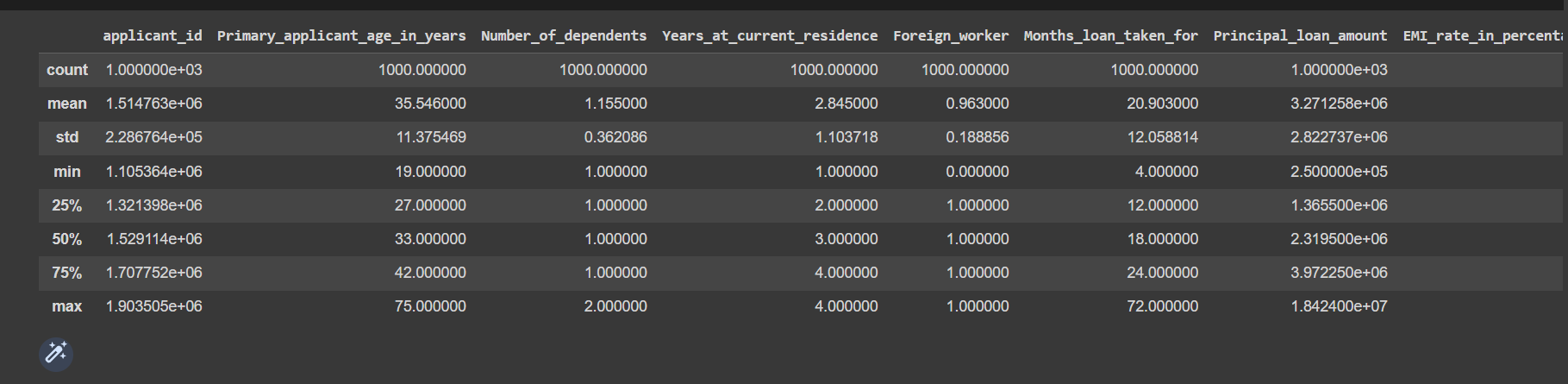
In this project we are going to be working on loan data set. Where we have two file one is with load data and other one is with applicant data now let us first import all the required libraires

Link to Task\_1 Collab: <https://colab.research.google.com/drive/1_pJhVZWnmDzZdVVrSv-2rOB4k2_ivHdI?usp=sharing>

* There are two data sets provided let us merge them using pandas
* Now, let us visualize the info of the data set

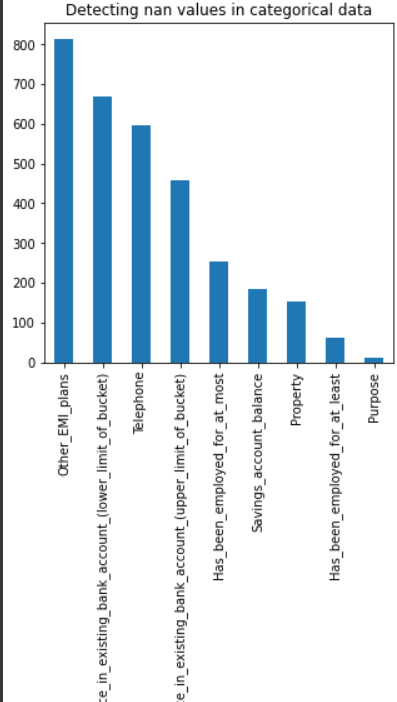


* Now, let us see the statistical numbers using describe function of pandas

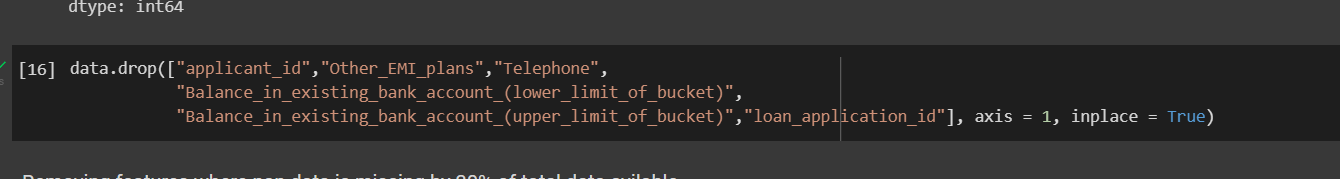


**1.Exploratory Data Analysis & insights.**

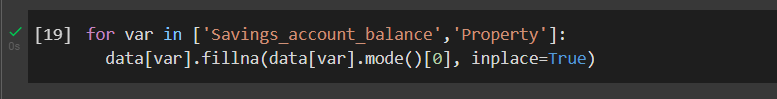
* There are Missing values present in the categorical data let us see how do we deal with it



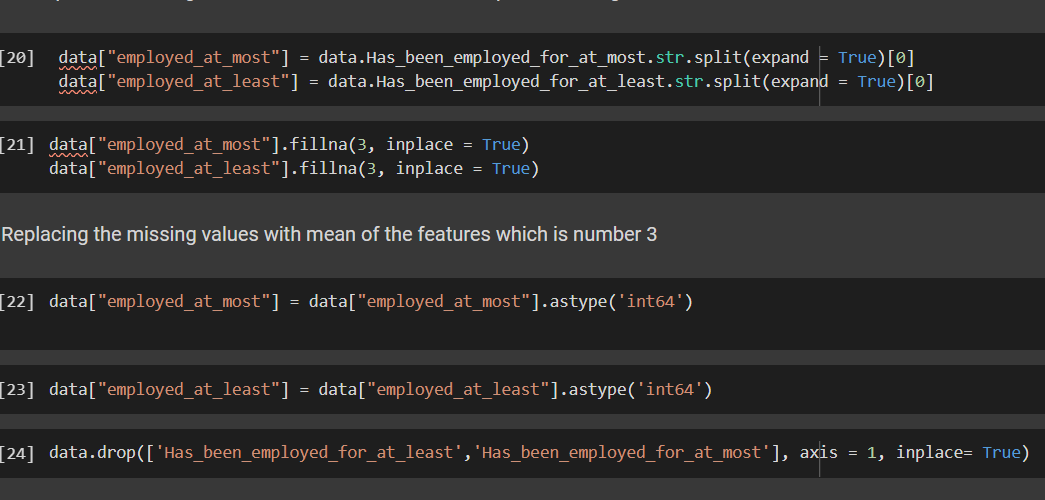
* There are few columns where the data is not available let us see which can we drop and which we can fill



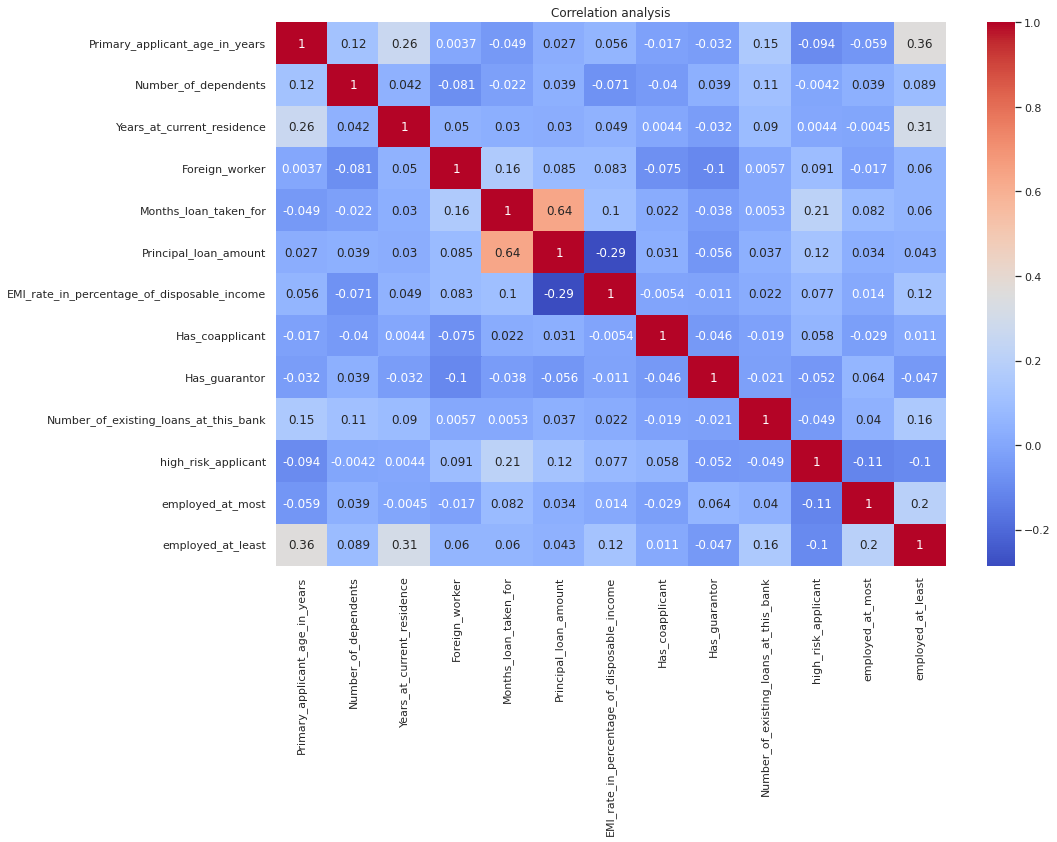
* Now let us drop few features where there the data is missing by more then 30% percent.



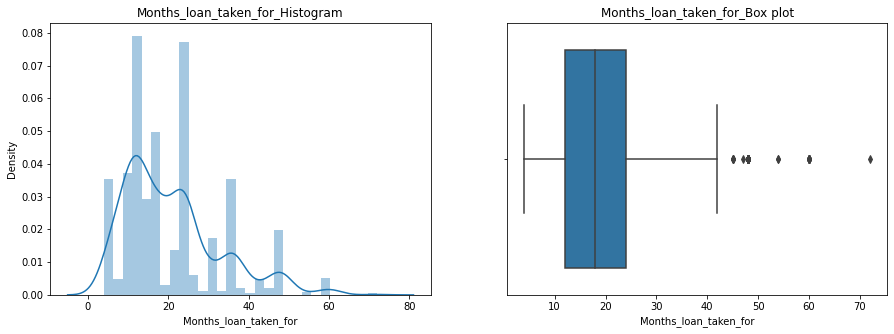
* Over here since it is string feature, I am going to replace the missing values with the mode of the feature column so that no data is lost

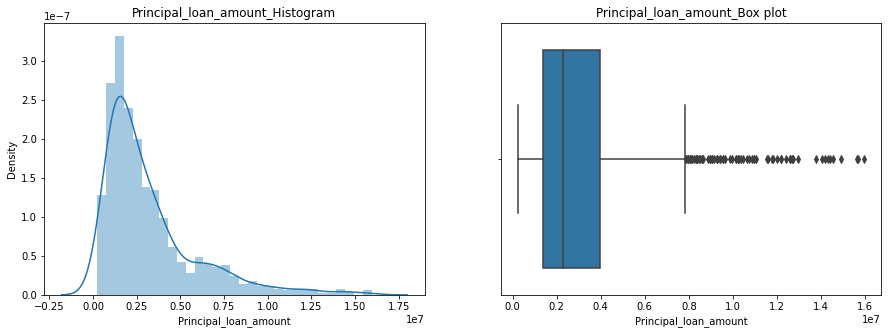


* Now as we see in this particular feature the data is in alpha numeric so let us strip and create new features so it can be useful in later in machine learning
* Since there is no missing values in the numeric data let us see corelation using seaborn and heatmap

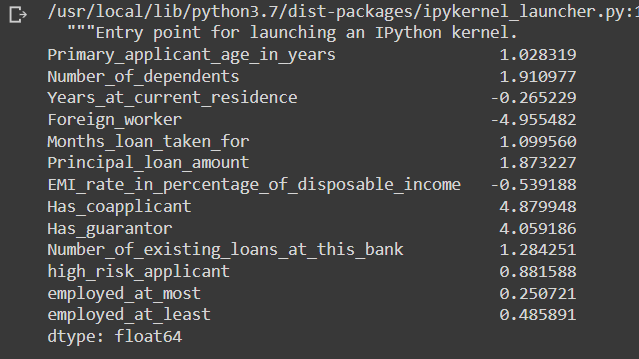


* Visualizing correlation between the different numerical data so that we can see if there is any partial values or relationship between the different columns
* Now let us see for skewness within the data in numeric columns and let us see which column we can normalize

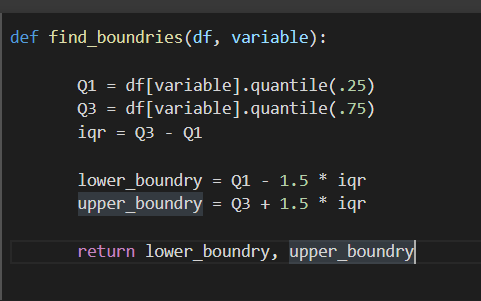




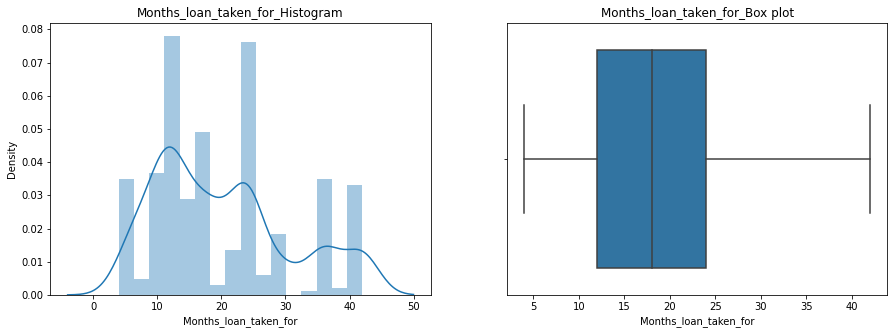
* So as be see there are outliers present in the feature called “Months\_loan\_taken\_for\_Histogram” and “Principal\_loan\_amount\_histogram” now let us normalize or remove the outliers present in the column
* Now let us create a helper function which will help us to find the upper boundary and the lower boundary of the features
* Meanwhile we do this let us visualize the skewness

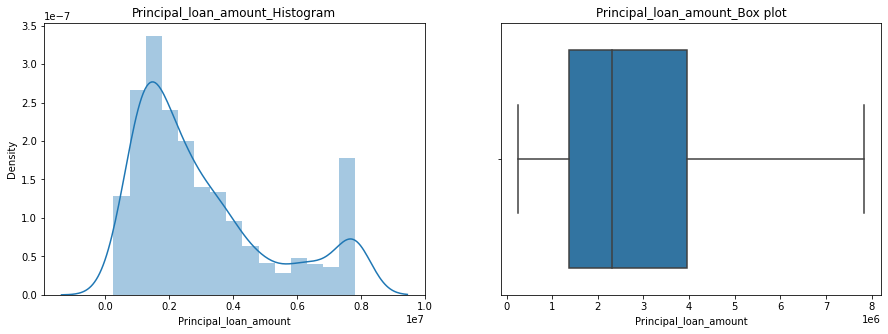


* So now as you sell most of the values are good and few are binary and what left is we are going to normalize



* This is a helper function using which we will normalize or replace the outliers once the outliers are removed you can see how the plots are visible in the down below

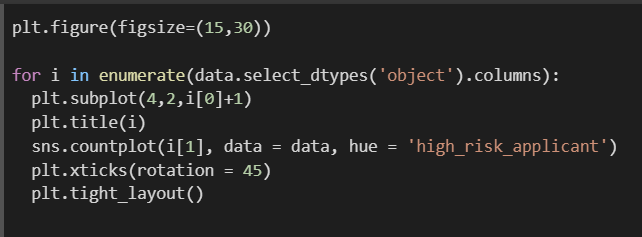


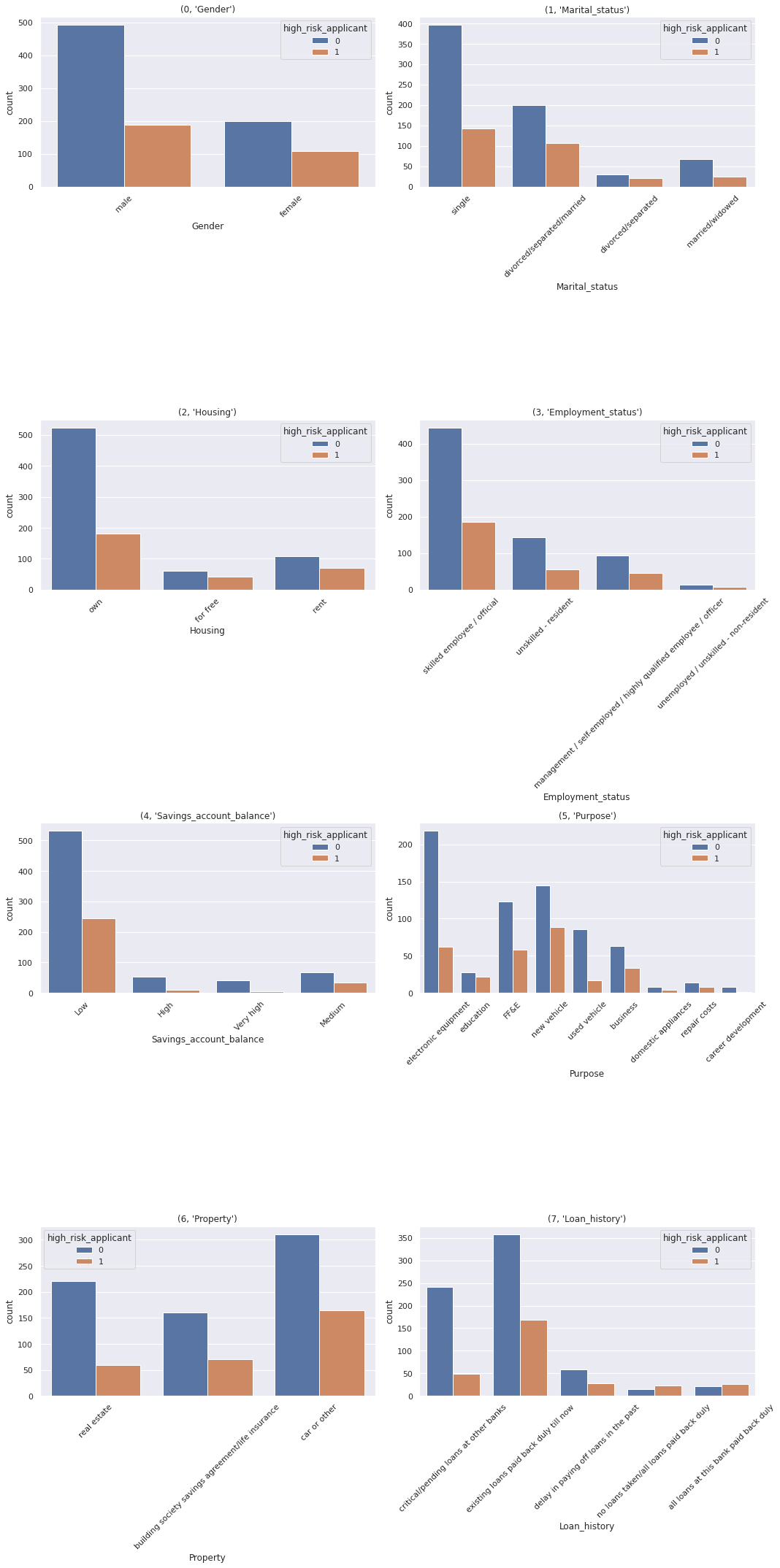


* Both the features are now normalized and the data is set be used in a machine learning project

1. **How can we segment customers based on their risk (of default).**

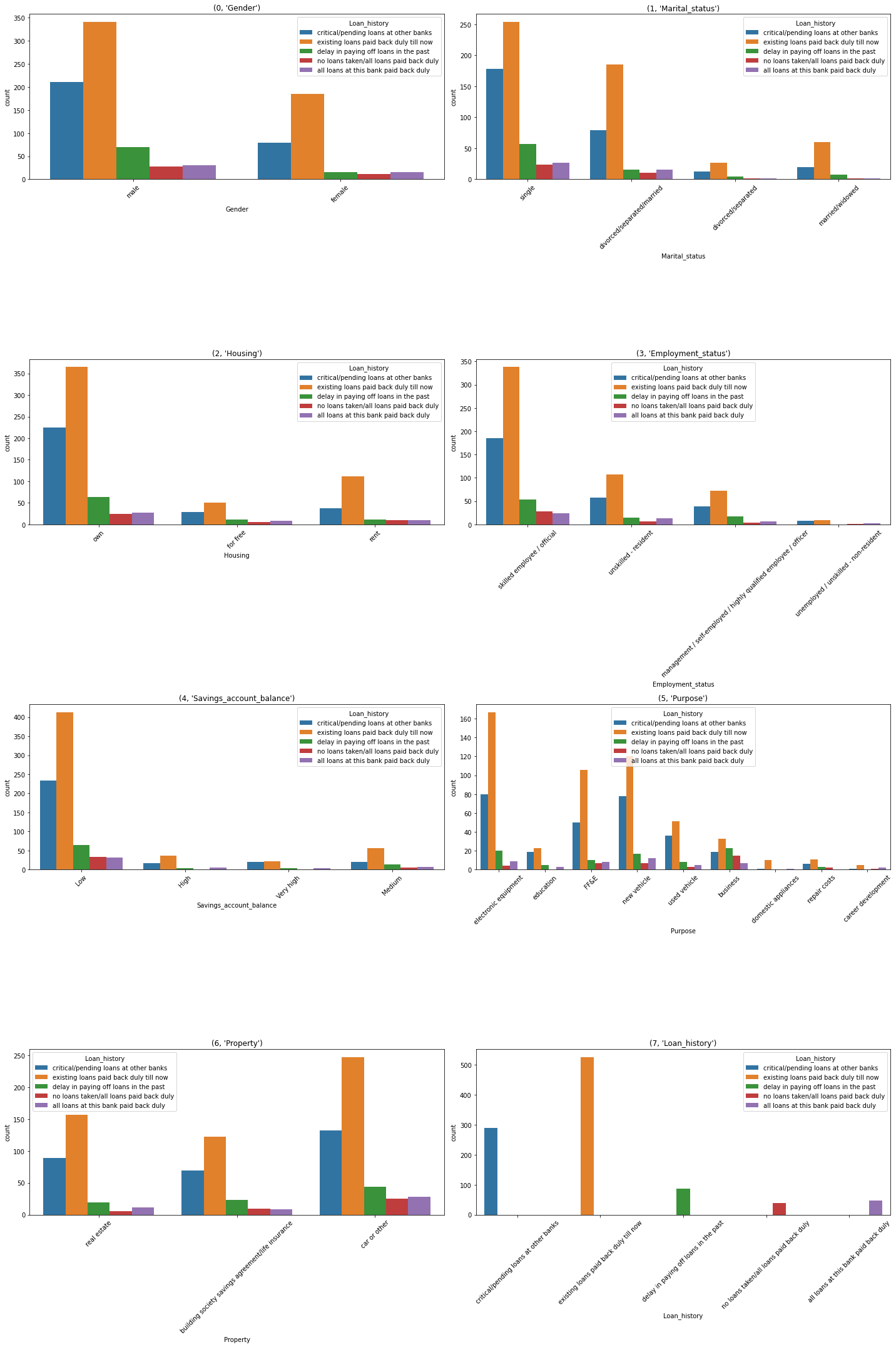
* Let us visualize all these categorical data using count plot and see how we can get insights and answer segment customers based on the output



****

* Will segment customerbased on gender, marital status, purpose and loan history

1. **Which of these segments / sub-segments would you propose be approved?**

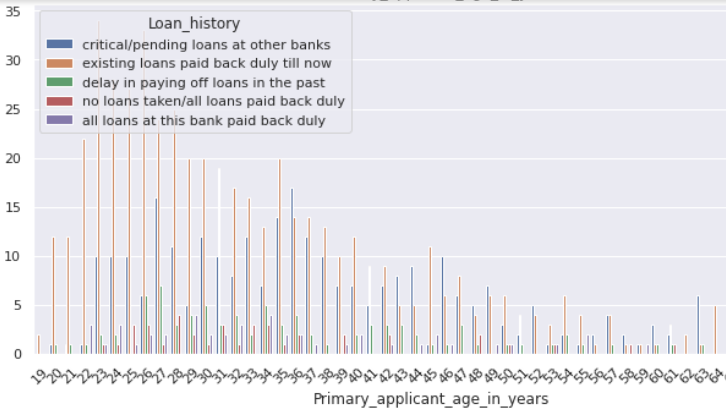
****

**Q. Would a person with critical credit history be more creditworthy?**

Ans. No, I don’t think that would be right as in the since he as critical credit history his credit must be monitored and subsequently notified on the trend of his account

**Q Are young people more creditworthy?**

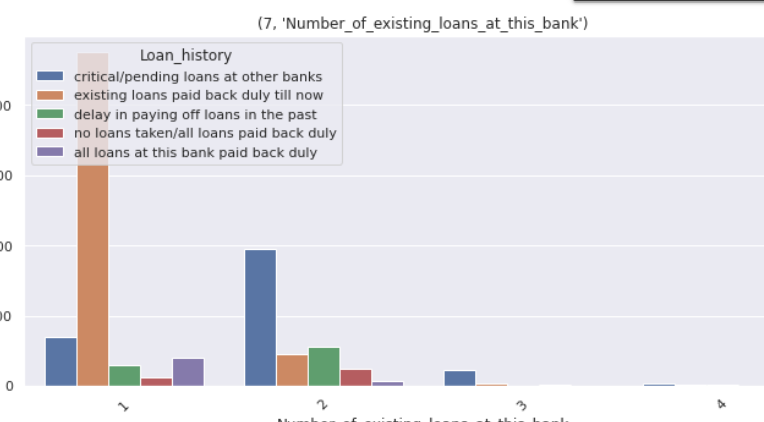
Ans. Yes, young people are more creditworthy but should check if they are employed or not



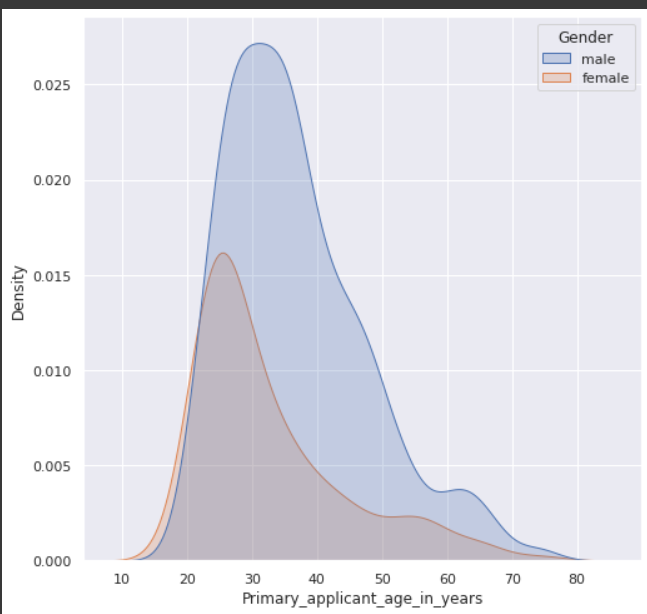
As we observe the range between 20 and 35 are more likely to pay back loans

**Q Would a person with more credit accounts be more creditworthy?**

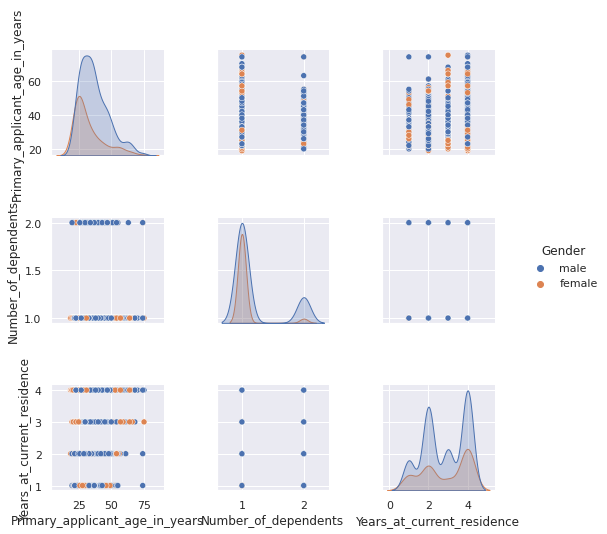
**Ans.** Depends upon the existing loans in other credit accounts and if he or her paying up or not.



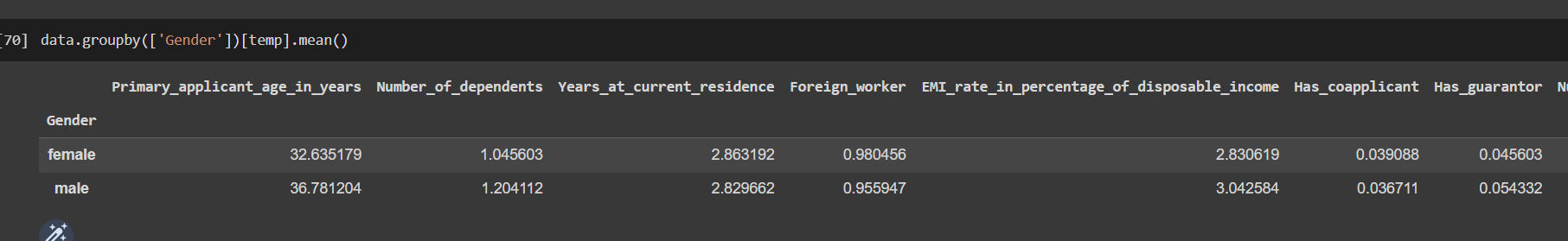
* Using KDE plot we see gender male have more density in the data compare to female



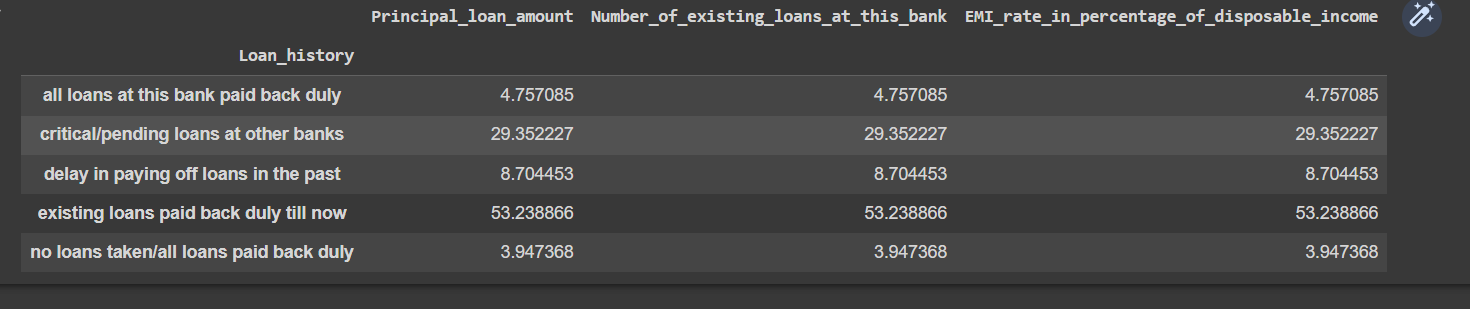
* Using pair plot to visualize the data as per three numeric features



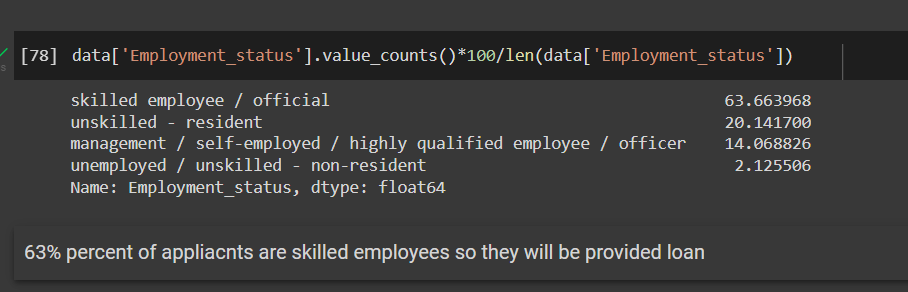
Grouping by the gender column with respect to all the numeric column we can understand the key differences between both the genders with respect to buying parity and more



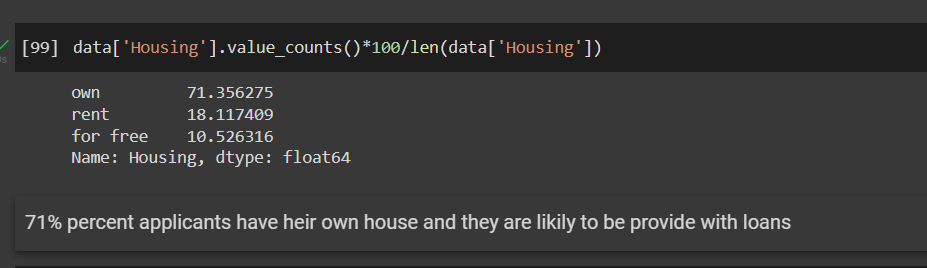
By using group by on loan history and relevant features we can see that with in the data set we see this trend where in more than 50% of the applicants have returned back the money which was bored and 29% have critical loan pending



63% percent of applicants are skilled employees so they will be provided loan



71% percent applicants have their own house and they are likely to be provide with loans



**Insights Plots above convey following things about the dataset:**

1. Loan Approval Status: About 2/3rd of applicants has been granted loan.

2. Gender: There are more Men than Women (approx. 3x)

3. Marital Status: 2/3rd of the applicant in the dataset is Marred; Married applicants are more likely to be granted loans.

4. Dependents: Majority of the applicant have 1 dependent and they are likely to accepted for loan. 5. Education: About 5/6th of the applicant is Graduate and graduates have higher proportion of loan approval

6. Employment: 63% of applicant is skilled employee and they have higher proportion of loan approval

7. 71% of the applicant own a house and they are likely to be provided with the loans

8. Applicant with credit history are far more likely to be accepted.

9. Buying Electronic equipment is the most common case when it comes to taking loans

10. Who are residing at a residence above 4 years they are having the greatest number of critical loans

11. applicant having two loans simultaneously are under critical loan under loan history

12. Among high-risk applicant women are more compared to men

Using Different Machine Learning Model to Predict Risk

Link to Task\_2 Collab: <https://colab.research.google.com/drive/1J47xFqOPQprrX1LWE_2J9L53EXXXiz1E?usp=sharing>

**Explain your intuition behind the features used for modelling**

During the EDA I have replaced or dropped all the values in the process and the features generated are being used now are the output of the due process

**Are you creating new derived features? If yes explain the intuition behind them.**

I have generated two features during the EDA and I tried of looking into more cells but the current features work good to the applied models.

**Are there missing values? If yes how you plan to handle it.**

**Yes,** there were missing values but I have delt with the missing values during the EDA part

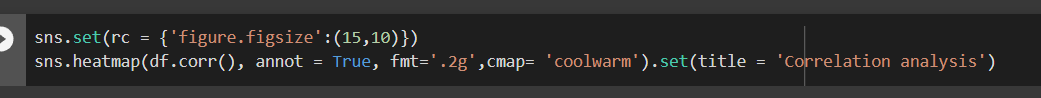
**How categorical features are handled for modelling.**

Will use one hot encoding to encode and then transform all the categorical cols and then create mother input and target and push it into the model

**So, Now let us start with the machine learning process**

**Models we would be working on**: **Deep Learning Conventional Model** **Random Forest Classifier, Decision tree, XGBoost Classifier, Liner regression model** and then save the best performing model

Let’s start



**Do you plan to drop the correlated feature? If yes then how.**

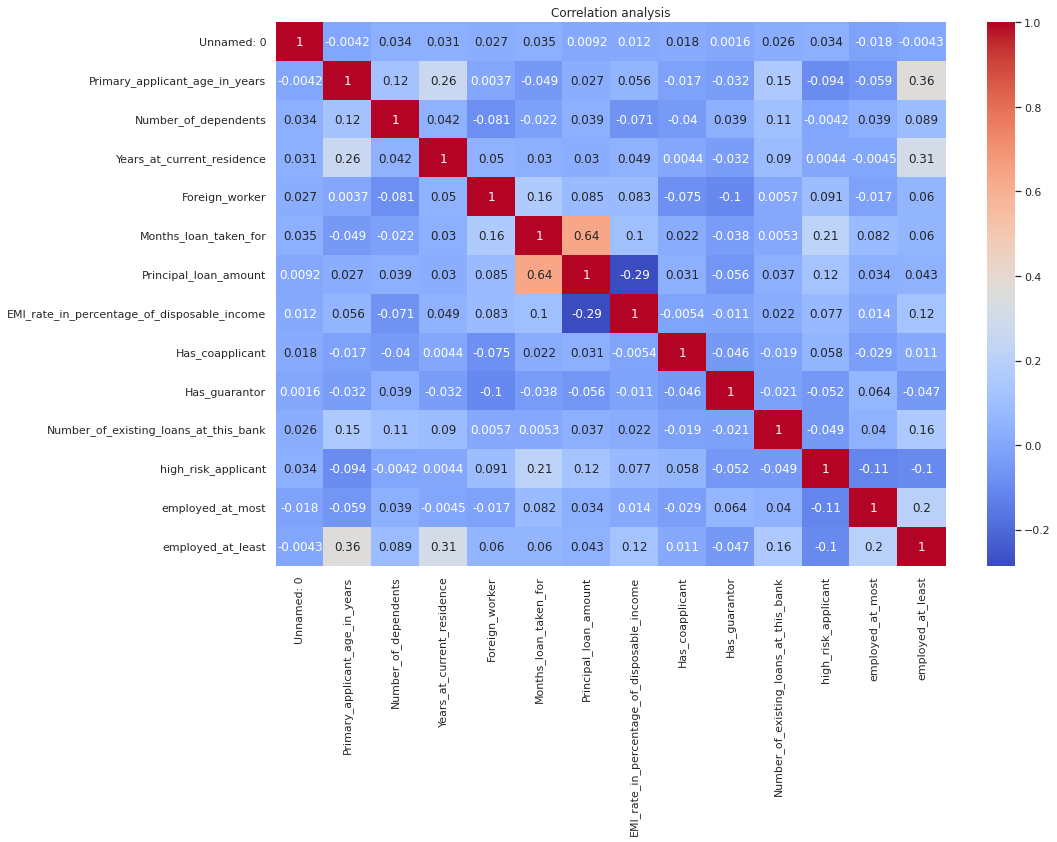
Over here we are going to use seaborn to create heat that is visible on the next page

While is see the corelation with respect to **the target column which is high\_risk\_applicant**

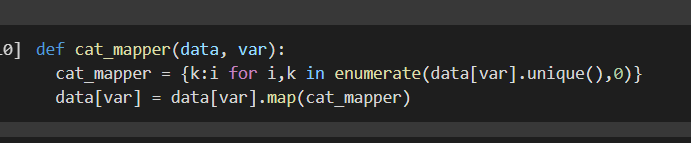
**I can’t find much corelation with target column to that extent that we need to delete the feature**

**Describe the features correlation using correlation matrix. Tell us about few correlated feature & share your understanding on why they are correlated.**

* **Applicant age** is corelated with the target column as in it directly impacts as the person when he is employed or gender is in range 20 to 40 we see a trend of loans that are taken is high
* **Emi percentage** is also corelated with target feature as in Emi to be played back is totally relatable to the risk of going into default so it is one of the main features
* **Months loan taken** do also impacts weather he would be able to pay back or not as in if his employment is at risk and many more corelated reasons



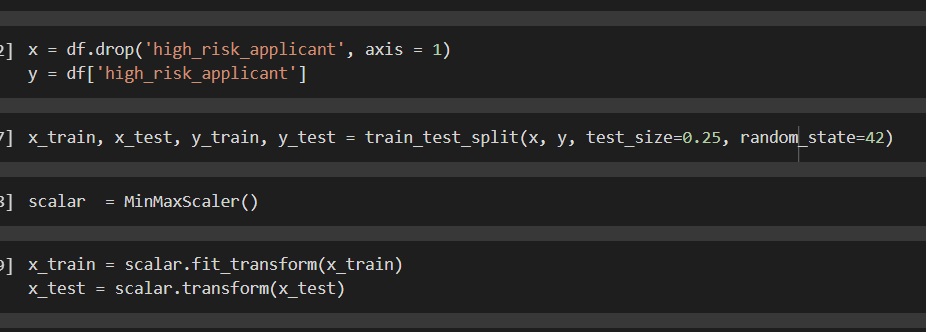
**Let’s convert all the categorical data to numeric using ordinal scaling using helper function that I have created**



**We Create two data sets stating x – inputs and y - targets**

**We split the data into train set and test set using train\_test\_split() of sklearn libraire**

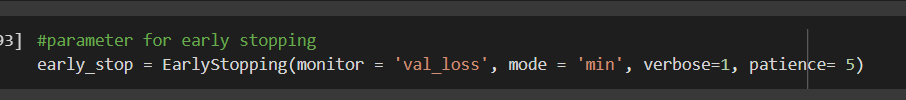
**Let’s** **fit and Transform the data using MinMaxscaler() so that values can be scaled from in the range of 0-1**



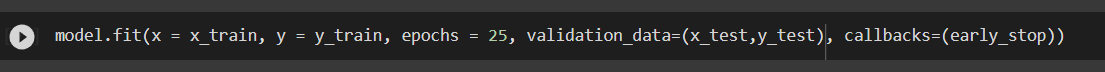
**Let us start with conventional model:**

* **Here I have provided 4 dense layers with neurons** of size 30,15,10,5 and final layer I have provided with 1 layer
* **I Have used Relu as** in activation function for all the layers
* **I have kept a drop out of 20% so that model doesn’t overfit**
* For the final layer I have used **sigmoid** activation to have better binary classification
* To compile the model, I have used loss function as in binary classification and optimizer to Adam as it is widely used for binary problems.

**Created parameter to stop early so that model doesn’t over fit**

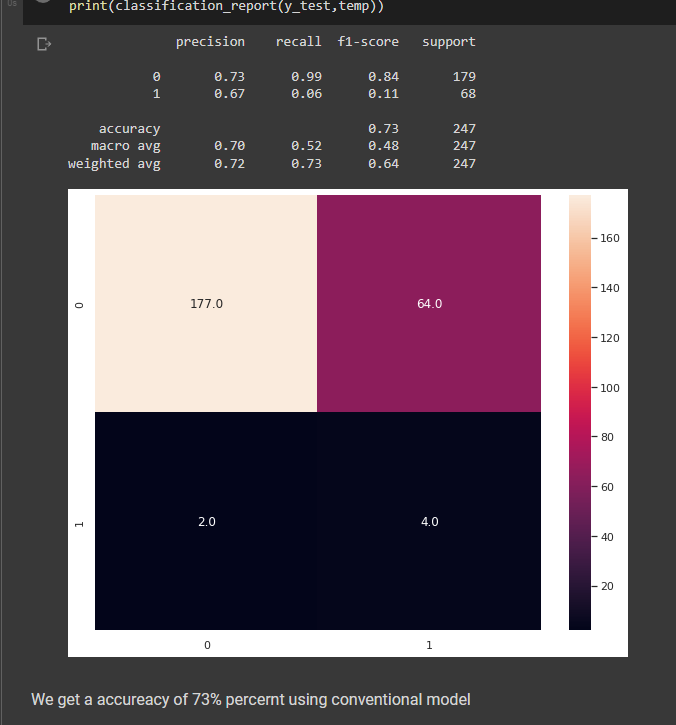


Let’s train the model on x\_train and y\_train and predict outputs on x\_test:

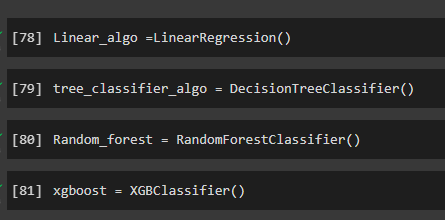


So once the model is fit we test on validation data x\_test we plot data on heat map with precision, recall and confusion matrix using sklearn.matrics

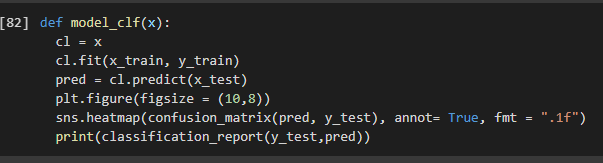
**Fitting conventions deep learning model**



**Let us now Import different models such as Random Forrest, decision tree classifier, xgboost classifier, liner regression and fit it to the inputs**

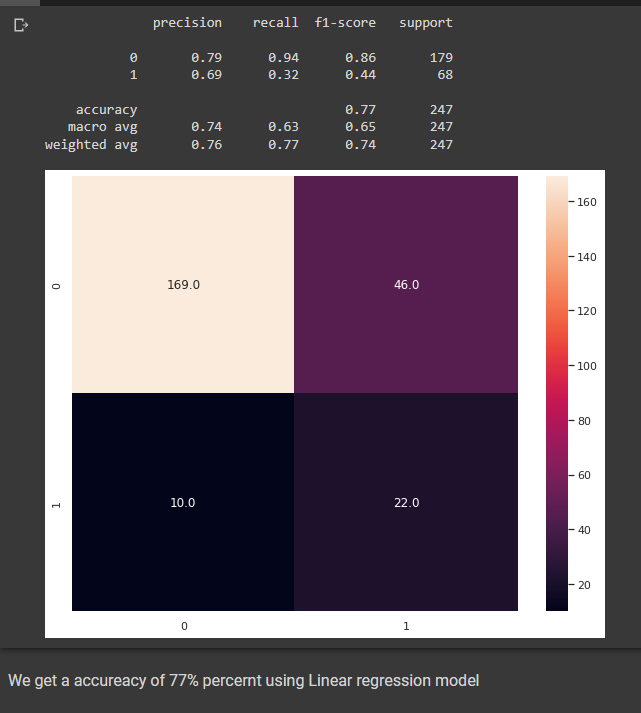


**Let us Now create a helper function which is used to call a model, fit a model, then visualize confusion matrix and there report parameters to track how all the models are performing**

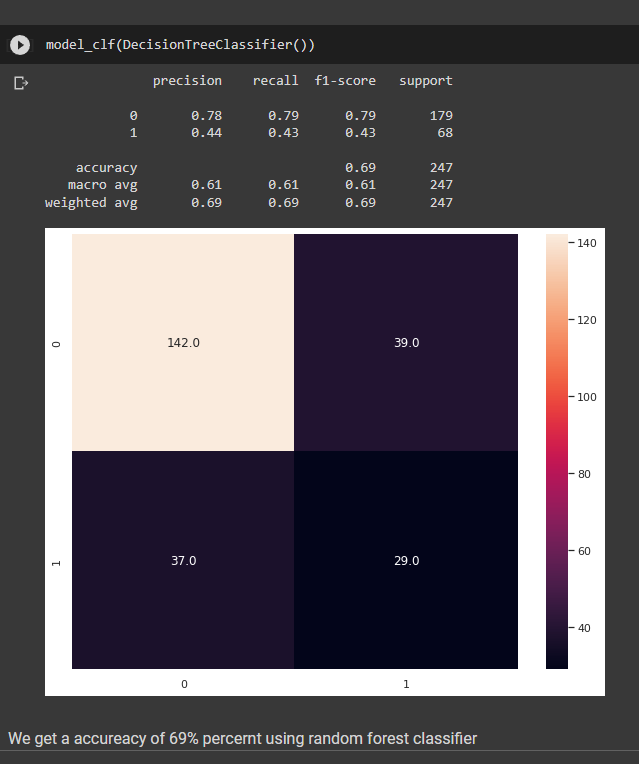


**Let us now implement this function to all of the models and see how they perform**

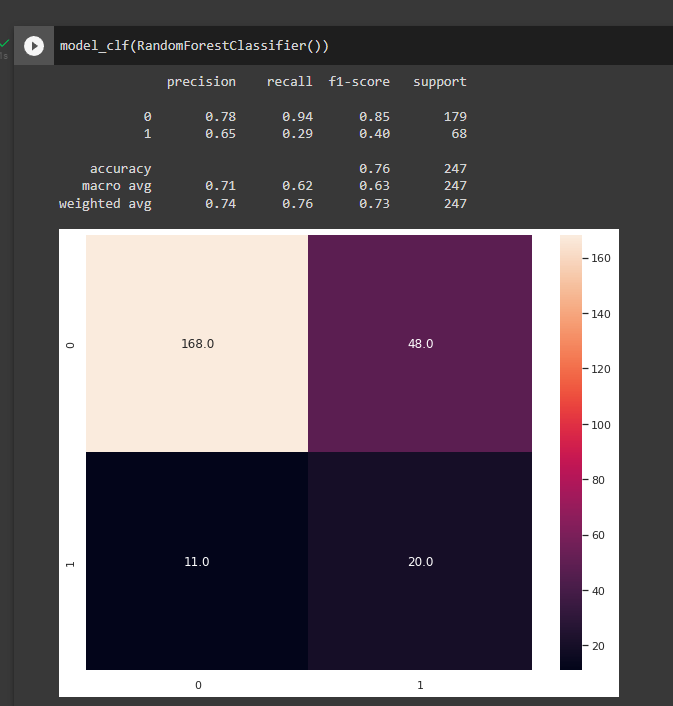
**Linear regression model**



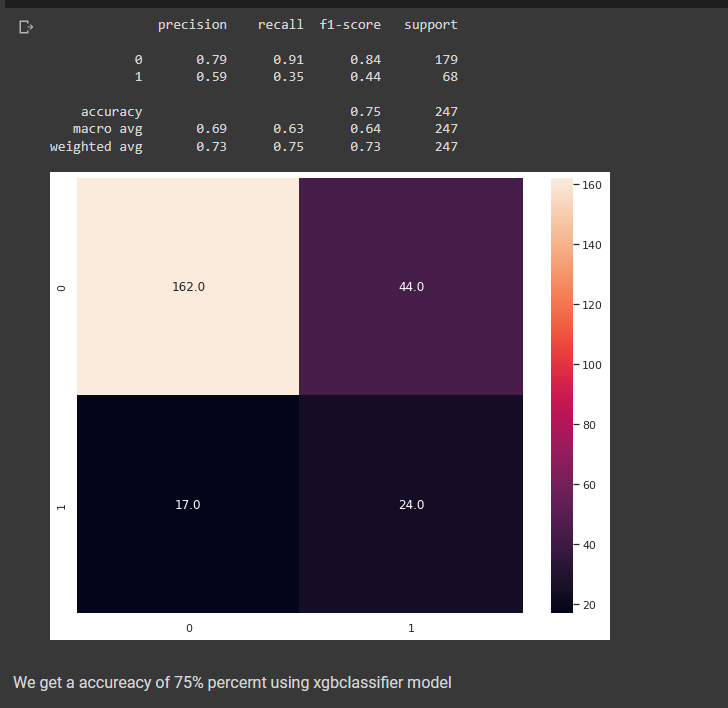
**Decision tree classifier model**



**Random Forest Classifier:**



**XGBclassifier:**

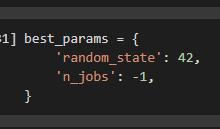


**As by Watching over all the models I suppose linear regression model is working with the highest accuracy let's try few hyper parameters to tune the other models and see if any of the model can work better than the linear model**

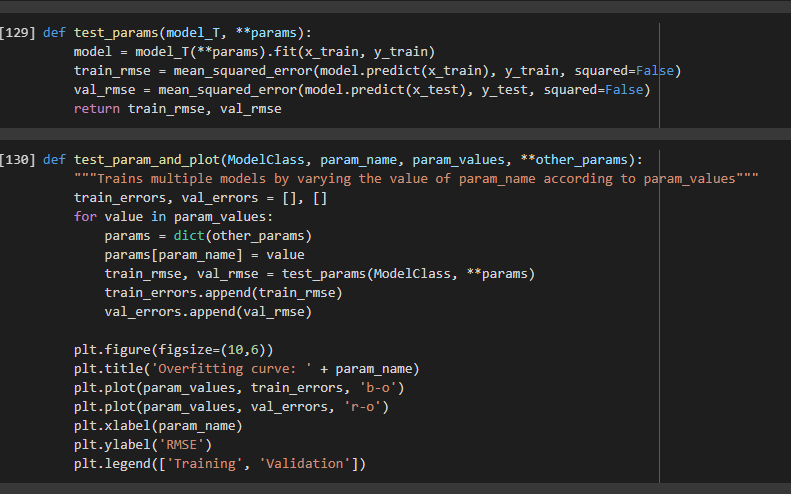
Let us start with the hyperparameter tunning wherein let us create two helper function

Where one is to test parameter and other one is to test and plot one parameter with various values and best parameters.

Let us now assume example of best parameters and test different parameters upon that



Basic parameter added to a dictionary



Let is now plot and test different parameters such n\_estimators , max\_depth, max\_feature

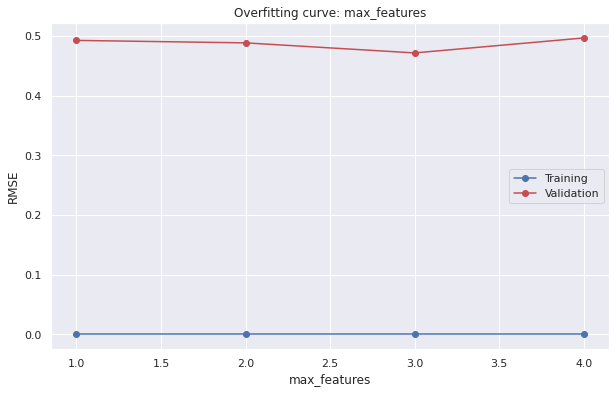
As I chose only those because they have the highest impacts on the score

n\_estimators are used to check number of forest

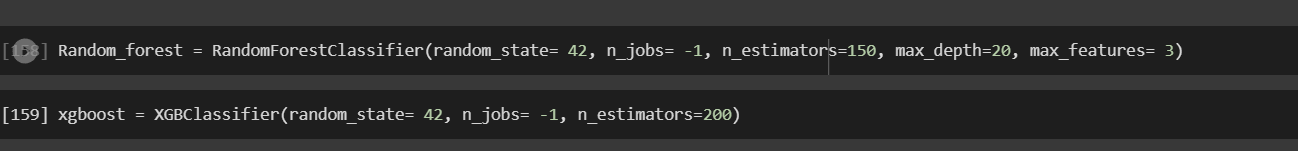
max\_depth to see depth of number of trees in a forest

Max\_feature is the number of features to consider each time to make the split decision

All the hyper parameter tuned are available on the collab notebook but for example I am attaching a tuned parameter on to the sheet

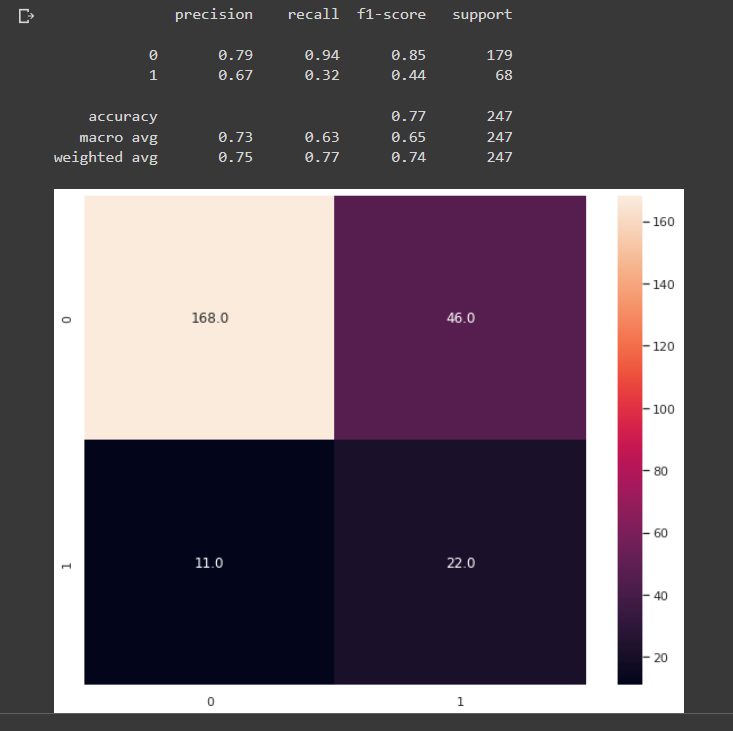


Once all the parameters that are best possible fit are known now, we will apply those into the models



Once again will plot the confusion matrix and check metrics and see how the model is performing

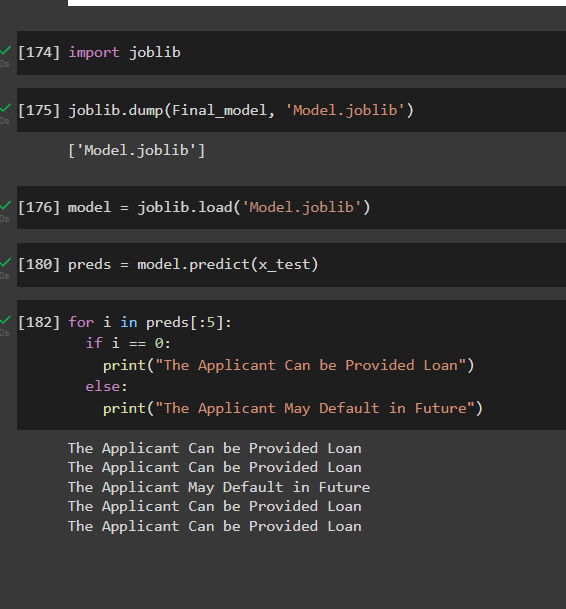
And I come to a conclusion that random forest is performing better



**Before it was 76 percent and now the model is improved by 1 percent and which takes the model accuracy to 77 percent**

When it comes to confusion matrix **true positive is performing good when compared to rest of the matrix but** False negatives is moderate in the matrix which could be managed by more tunning and by creating new features

Will save the model using job lib



Summary:

Considering this work, the key messages and conclusion of this work could be summarised as follows

1. The project results show that the Random Forest performs better then rest of the models on the training and validation datasets.

2. The data was skewd. we have normalize the data by removing the outliers present and used log on the target set to normalize it.

3. The model improved by implementing hyperparameter tunning as the score improves eventually once it is applied

4. Creating new features so that the model performance can be improved

5. Encoding and transforming data with the help of ordinal encoder so that we convert the categorical data into numerical and feed it to the input inturn it can be fitted to the desired model

6. we split the data and assign train inputs train targets and validation inputs and validation targets

7. Once we fit and tune the model with different parameters, we finalize the model and fit it to the tuned parameters and then we predict the output to the test data we had split in the process