**Introduction:**

This part of the project contains the linear regression, where we try to find the loan amount an applicant could receive.

**Preprocessing:**

Feature Selection and Engineering:

1. Income
2. #CreditCards
3. #LoanAccounts

No feature engineering was conducted.

Since the data is not of desired size, most of the data was retained and only the null values were removed.

This step brought the mean absolute percentage error(the error measurement technique used) from an average of 90% to less than 50%.

**EDA:**

1. We see that the largest amounts of loan are given to the middle class income, this shows that the higher income don’t require large amounts as the can probably compensate for the remaining by themselves. The lower income are most probably not awarded a larger amounts in loan due to a lack of collateral for the bank.
2. The second graph shows that middle class earners are in the largest need of loans, with negligible people taking loans from the extremes of the income spectrum, though the reason for both ends migth be different. Higher income families would rarely require such financial aid while the lower income would find it difficult to repay it, and are probably just rejected during application stage.
3. Here we come across to find an incredibly less amount of people having two credit cards when compared to single card owners. We can also see that on average single card owner have a higher loan requirement, comparing to dual card owners, showing they have higher income.
4. In the fourth graph the shape we see (thick in the middle and thinner at the extremities) is the same outcome as the second graph. But it is interesting is that the 2 credit card holders show a similar pattern, which means that they go against our interpretation of the third graph that 2 card owners have higher income in general.

**Project Summary:**

The project was standard, the data was split using the inbuilt sklearn function train\_test\_split().

Then a linear regression model was made and trained using the train set received from train\_test\_split(). The trained function was made to run the test inputs. The result was stored in a separate variable ‘Y\_Pred’. Since this is a linear regression, to calculate the accuracy, we calculated the mean of the percentage error between the respective values of ‘Y\_Test’ and ‘Y\_Pred’.

We printed the accuracy as (1-(mean\_absolute\_percentage\_error)). The average accuracy was found to be around 60%-70% though we could have had a better accuracy with a larger data set. The large variance in the accuracy can be attributed to the small database.