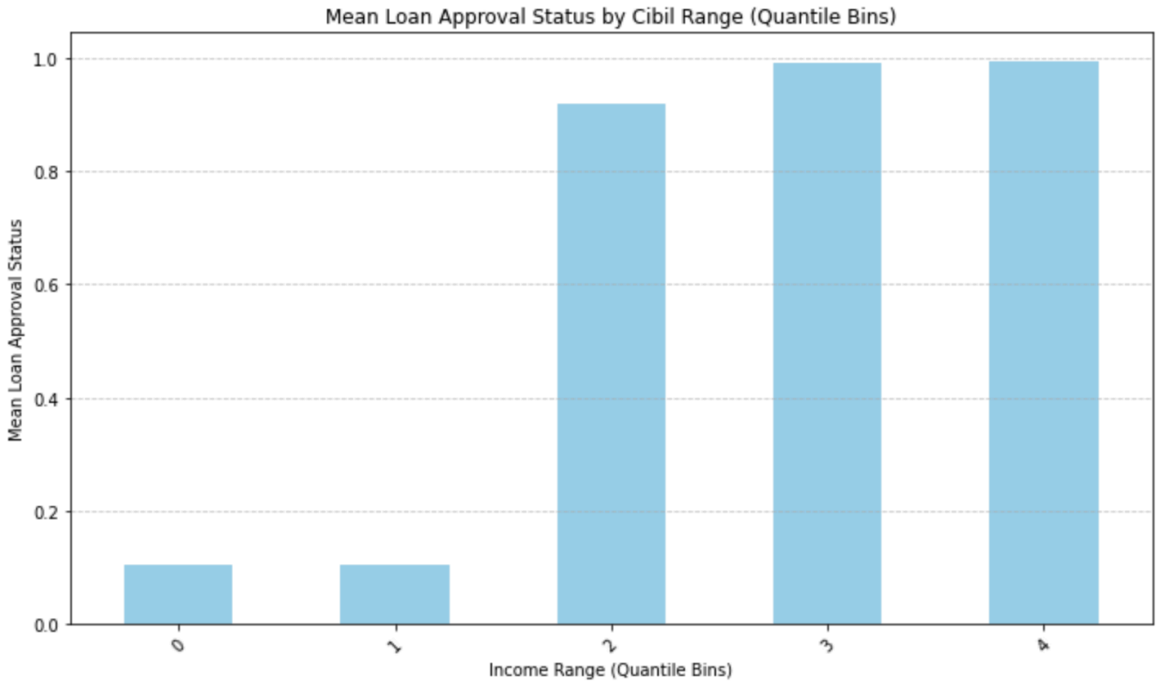
SPINNY ASSIGNMENT

EDA

4269 Rows with 14 Columns and No Null Values

Education: Ordinal Data Type , Replaced 8th,10th,12th,Graduate with 1,2,3,4 respectively
Employment_Status: One Hot Encoding with Columns Salaried,Business and Freelancer
Self-Employed: Dropped this columns as it is orthogonal to Salaried Column
Loan_Status:Converted Approved:1 and Rejected:0

Out Of all the Features , Only **Cibil Score** was Correlated with Loan Status (0.77)



Base Model :
Predicting Cibil_Cohorts <2 with 0 and Cibil_Cohorts>=2 with 1 gives -

- Accuracy: 0.939
- Precision: 0.968
- Recall: 0.932

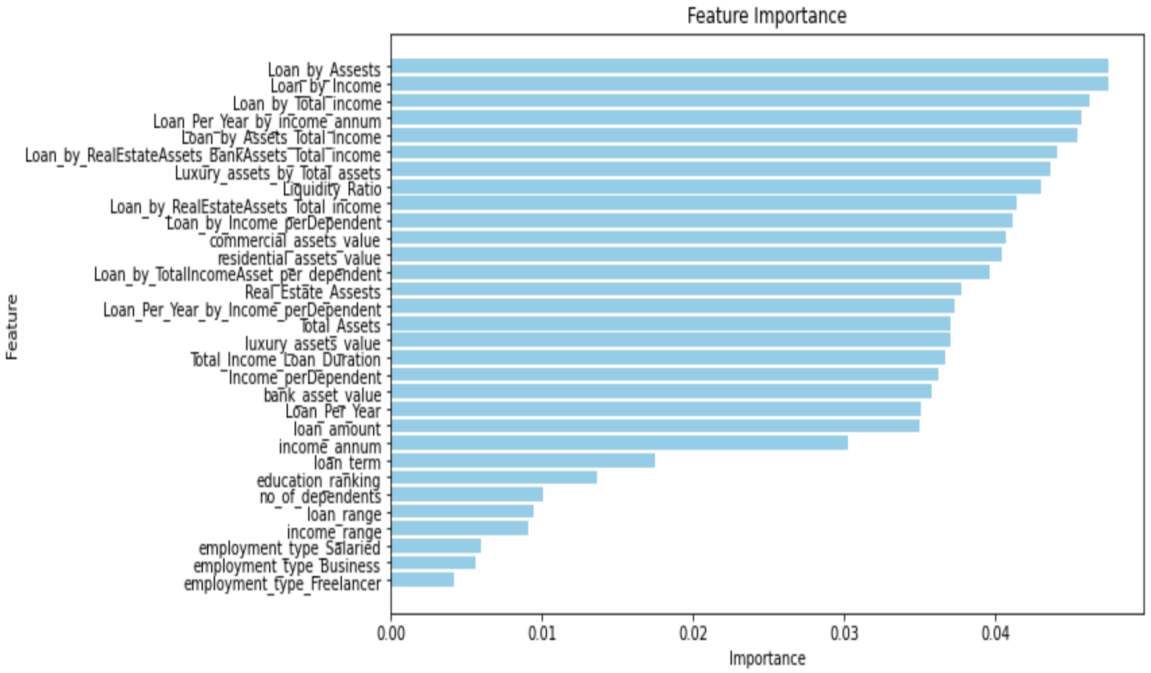
Feature Engineering

Risk attached to Loan Requested - Loan Amount individually cannot signify the likelihood of Loan Approval but the Ratio of Loan Amount to Income and Assets can be a good indicator . A Loan Of High Amount can be approved only to a person with Good Income and Assets not to a Low Income or Low Assets person.
Normalisation with Number Of Dependents - Income and Assets should be normalised with number of dependents (*Added +1 in Denominator to handle 0 Dependents*)
Financials Profiling - Users can be profiled by Total Assets in Real Estate , How Much Liquid Assets they have and How much they Spend On Luxury Assets

Created these Features Using the above 3 Strategies -

1. Loan by Income: Ratio of loan amount to annual income.
2. Income per Dependent: Adjusted annual income per dependent.
3. Loan by Income per Dependent: Ratio of loan amount to adjusted income per dependent.
4. Loan Per Year: Loan amount divided by loan term (assuming loan term is in years).
5. Loan Per Year by Income: Ratio of loan amount per year to annual income.
6. Loan Per Year by Income per Dependent: Ratio of loan amount per year to adjusted income per dependent.
7. Total Assets: Sum of all types of assets.
8. Real Estate Assets: Sum of residential and commercial assets.
9. Loan by Assets: Ratio of loan amount to total assets.
10. Total Income Loan Duration: Total income over the loan term.
11. Loan by Total Income: Ratio of loan amount to total income over the loan term.
12. Loan by Assets & Total Income: Ratio of loan amount to the sum of total income over the loan term and total assets.
13. Loan by Real Estate Assets & Total Income: Ratio of loan amount to the sum of total income over the loan term and real estate assets.
14. Loan by Real Estate Assets & Bank Assets & Total Income: Ratio of loan amount to the sum of total income over the loan term, real estate assets, and bank assets.
15. Loan by Total Income & Asset per Dependent: Ratio of loan amount multiplied by the number of dependents plus one to the sum of total assets and total income over the loan term.
16. Luxury Assets by Total Assets: Ratio of luxury assets value to total assets.
17. Liquidity Ratio: Ratio of bank asset value to total assets.

Used Random Forest Classifier to Find the Most Important Features after CIBIL Score -



Inflection Point was 0.03 . Dropped all Features with Importance less than 0.03

MODELLING

Used **Cibil Score** and other **Important Features** to Predict the Likelihood of Loan Approval (TARGET VARIABLE : Loan Status) using **Logistic Regression** And **Random Forest** Classifier . For Hyper Parameter Tuning , Used **Grid Search CV** and selected the best Classifier in each Algorithm . Calculated Precision , Recall and Accuracy as Evaluation Metrics .

In this Scenario Both Precision and Recall are Important as we dont want defaulters and also we would like to give loans to as many people as possible. But if the Budget to give Loans is fixed then Precision > Recall.

Results

RF

```
Best Parameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
Cross-Validation Mean Accuracy: 0.9967789165446559
Precision: 1.0
Recall: 1.0
Accuracy: 1.0
```

Logistic Regression

```
Best Parameters: {'C': 0.1, 'class_weight': None, 'max_iter': 100, 'penalty': 'l1', 'solver': 'liblinear'}
Cross-Validation Mean Accuracy: 0.9376281112737921
Precision: 0.949814126394052
Recall: 0.9533582089552238
Accuracy: 0.9391100702576113
```

Random Forest Classifier gave the best Result . Even tested it on all Slices of Data i.e. all Cibil_Score Cohorts and it outperformed in all scenarios

Created an Inference Code that will Predict ‘Approved’ or ‘Rejected’ based on any sample Input using the best Random Forest Classification Model