**Bureau\_Assignment Report**

Submission by-: Kishan payadi

Collab link-: <https://colab.research.google.com/drive/1Q2yKQ9t26VxxZeeERWF-guJiHZgttNSZ?usp=sharing>

Github link-: <https://github.com/kishan2k2/Bureau_Assignment>

**Problem Statement**

Assume you are a loan risk officer at a large bank and you are tasked with determining whether a two-wheeler loan application will be accepted or rejected based on the data shared by the loan applicant and some additional data extracted about them from 3rd party sources.

There are 3 files as part of this assignment:

1. Assignment\_Train.csv - This file contains labelled data containing all relevant information along with the *“Application Status”* variable that needs to be classified.
2. Assignment\_Test.csv - This file contains non-labelled data with the same information as present in the training set but without the *“Application Status”* variable.
3. Assignment\_FeatureDictionary.xlsx - This file contains details and brief description about individual variables present in the training and test files.

Use the UID columns as your unique identifier for each row.

**Approach taken.**

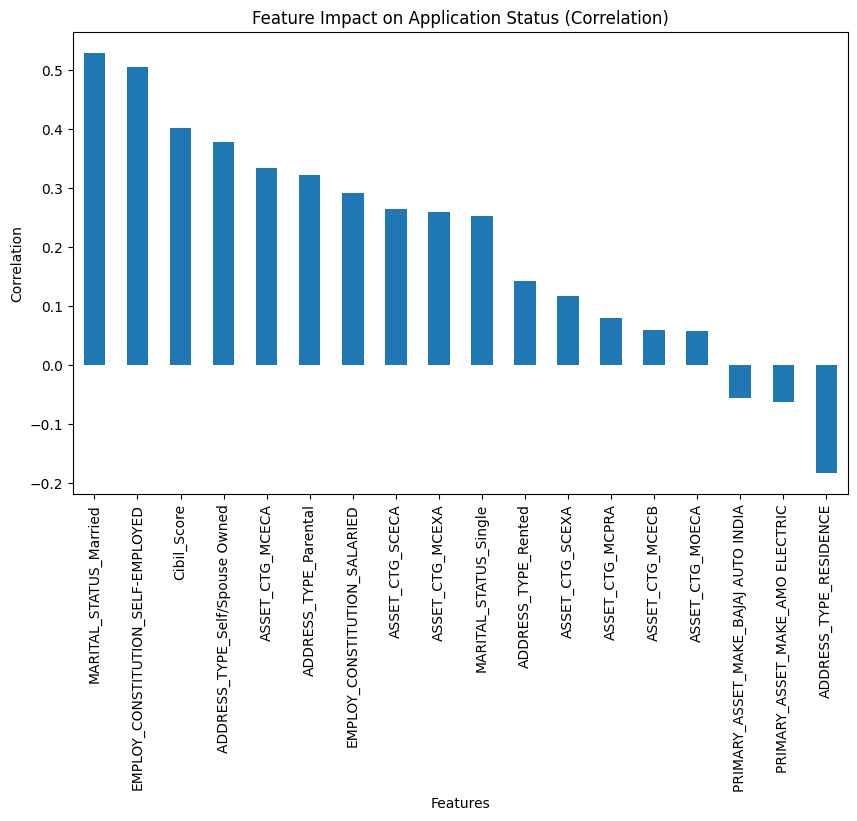
1. Data cleaning.
2. EDA.
   1. Insights and conclusion of data.
3. Feature engineering.
   1. Imputation using KNN and simple imputation like mean.
   2. Dimensionality reduction using PCA features.
4. Model building.
5. Model Selection.
6. Hyper Parameter tuning.
7. Model evaluation.
8. Generation of prediction.csv

**Data Cleaning.**

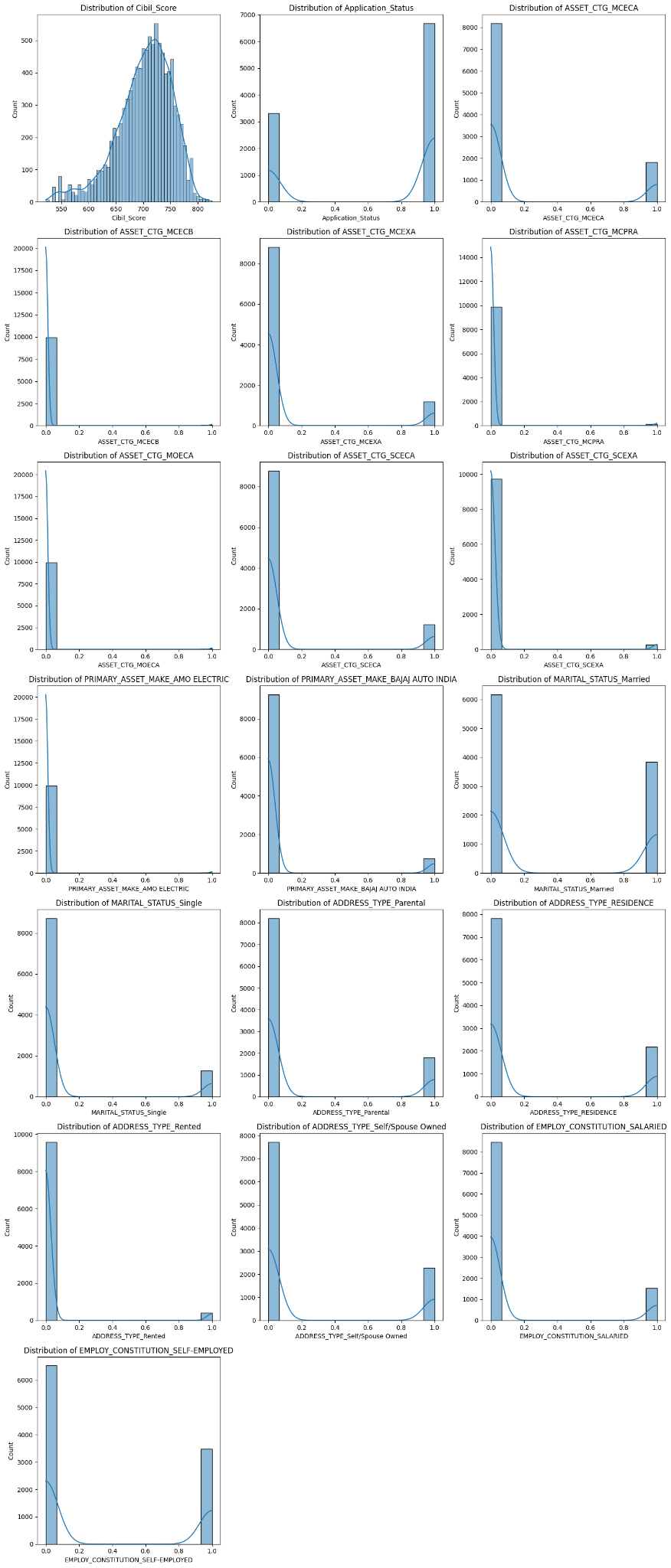
1. Important features like cibilScore asset categories, have a lot of missing data.
   1. Around 50% so it won’t be wise to use simple imputation like mean or dropping out or any other statistical method.
   2. Will have to use advanced techniques like using the nearest neighbor value based on Euclidean distance.
2. A lot of irrelevant data in the data which adds extra dimensionality causing problems because of the curse of dimensionality.
   1. Removing primary keys and redundant data like 'FIRST\_NAME', 'MIDDLE\_NAME', 'LAST\_NAME', 'mobile', 'DEALER\_NAME', 'DOB', 'EMPLOYER\_NAME', 'Pan\_Name', 'name', 'APPLICATION\_LOGIN\_DATE', 'AADHAR\_VERIFIED', 'HDB\_BRANCH\_NAME', 'Primary\_Asset\_Model\_No', 'Personal\_Email\_Address', 'upi\_name', 'EMPLOYER\_TYPE', 'MOBILE\_VERIFICATION'
   2. Concatenating the correlated data, i.e. the column name starting with Phone\_Social\_Premium.
3. There are many numerical data which are given as objects so fixing them.
   1. Cibil score, asset cost, phone digital age, phone name match score.
4. Handling categorical columns.
   1. One hot encoding them.
5. Imputation.
   1. Simple imputation-: This gave an accuracy of over 94%, but it was not robust so dropping it.
   2. KNN-: Using the remaining data to impute the missing data based on 5 nearest neighbors. This reduced the accuracy of the model but it was robust and consistent throughout all models.
6. Dimensionality reduction.
   1. One hot encoding of the data increased it to 127 columns 🤯
   2. Need to reduce it.
      1. Setting a threshold of 0.05 on the correlation between feature and target, and removing if it is less than that.
      2. PCA It gave the same result as the first method and it adds unexplainability and it is sensitive to outliers so dropped it.
   3. Reduced columns to 18 based on thresholding method.

**EDA, insights and conclusion.**

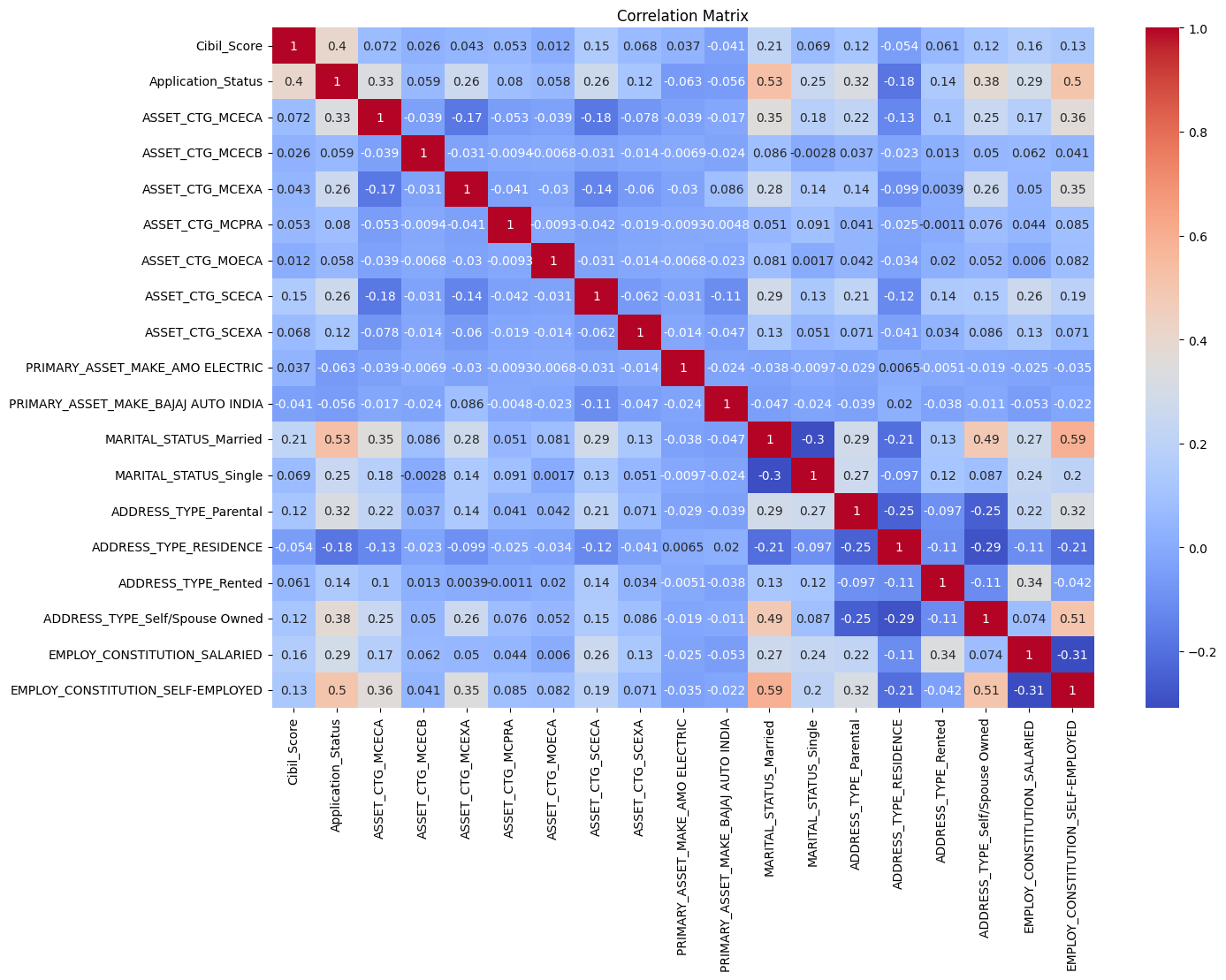
1. Correlation of features with application status {Check the notebook if not clear}.
   1. Being married, self employed, good cibil score, owning a house increases your chances loan aproval.



1. Imbalance in the data.

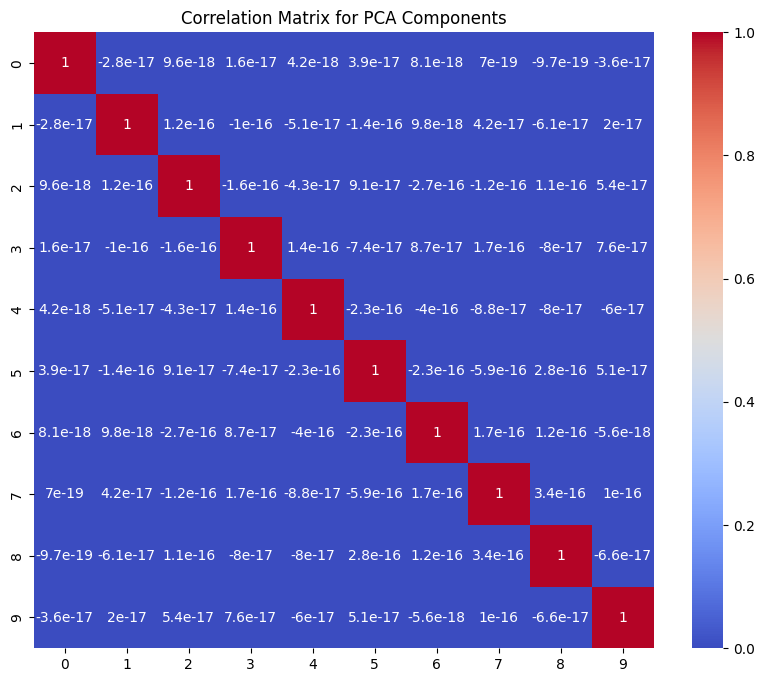
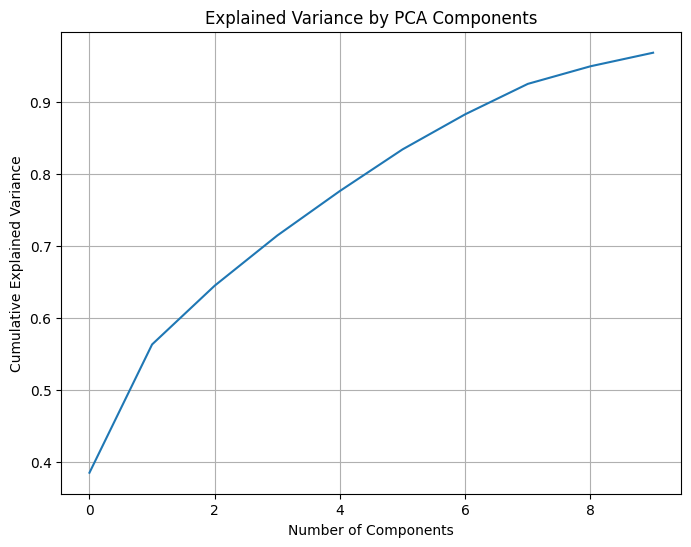
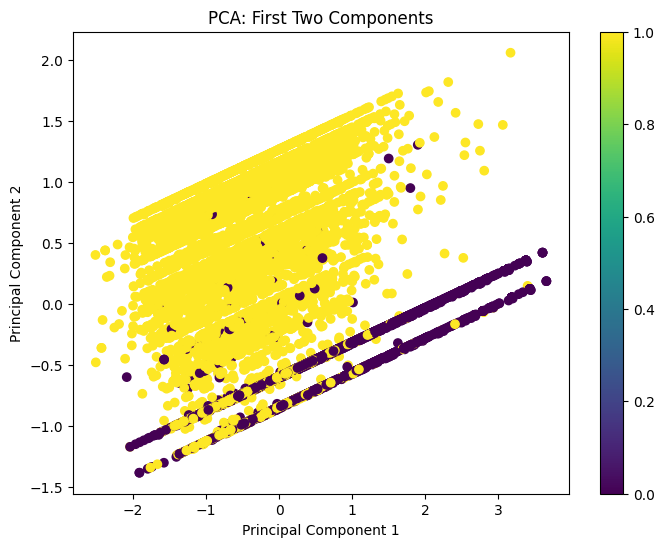


1. Correlation matrix between all the features, most of the diagonals are 1 and remaining are between -0.3 to +0.3 More improvements can be made..



**PCA and Dimensionality reduction.**

1. We can reduce more features and have less correlated features using PCA.
2. Replacing all the features to 10 dimensions retaining 95% of the variance.
3. The correlation matrix between pca components is ideal i.e. diagonal are one and all remaining are near zero.

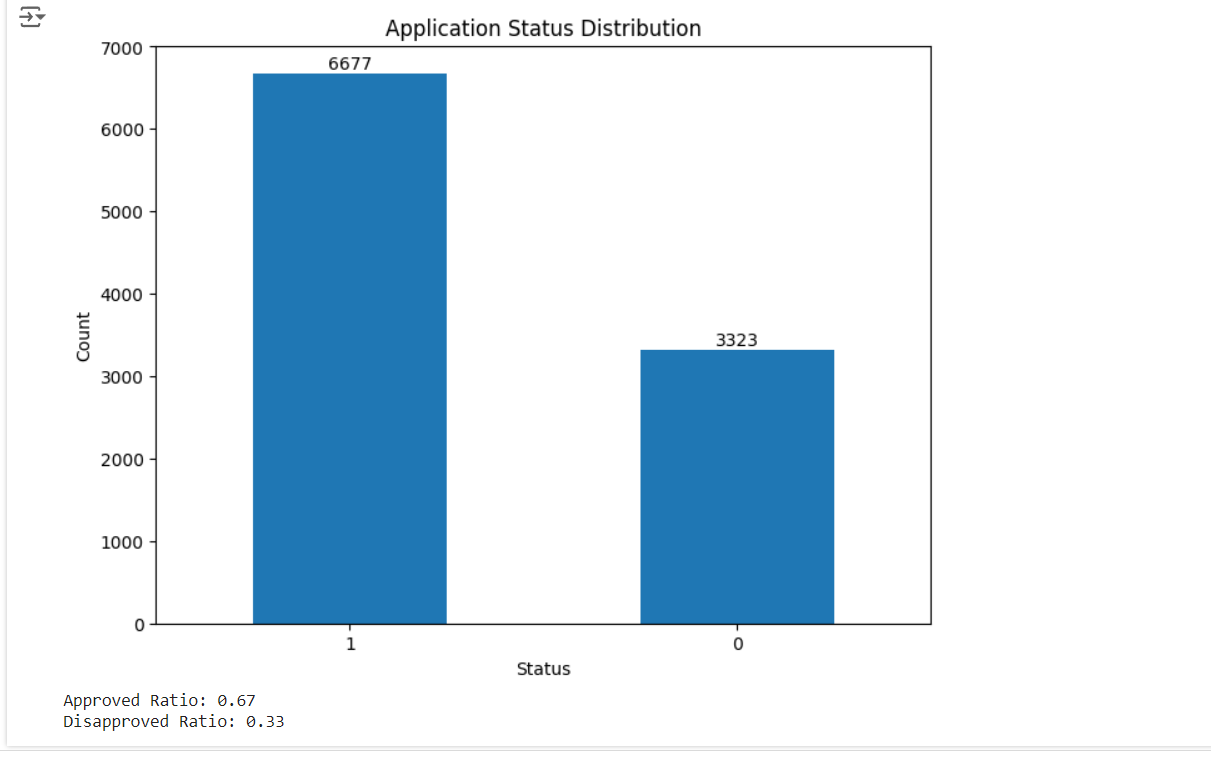


**Building model and Performance on a train data set**

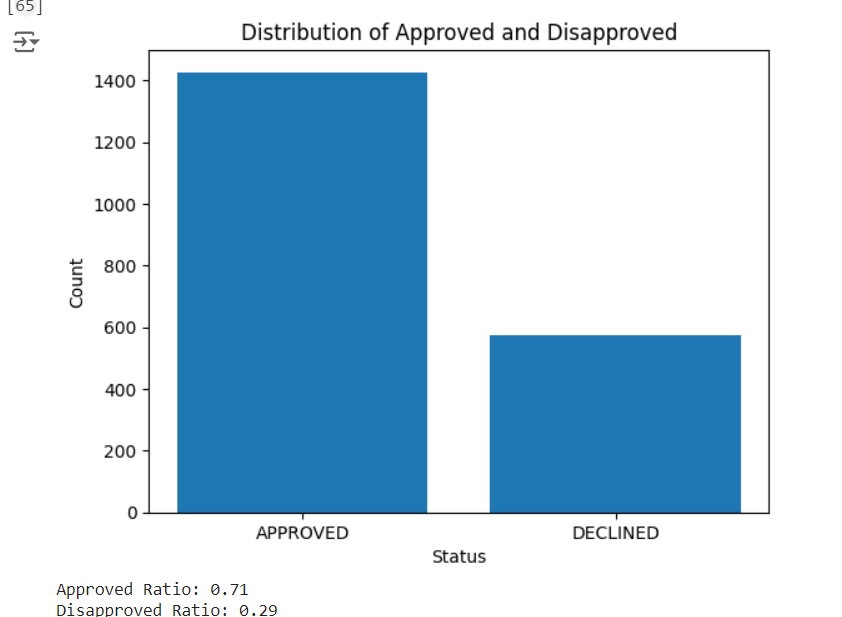
|  |  |  |  |
| --- | --- | --- | --- |
| Model | Accuracy | Precision | F1-Score |
| Logistic Regresion | 83 | 85 | 83 |
| Decision Tree | 81 | 82 | 81 |
| Random Forest | 82 | 83 | 82 |
| Gradient boosting. | 83 | 85 | 83 |
| Gradient boosting + hyperparmeter tuning. | 84 | 84 | 84 |

**Final Classification report**

1. Training data.

****

1. Testing data it is very much close to the training data **😁.**

****