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.
    a. Insights and conclusion of data.
3. Feature engineering.
    a. Imputation using KNN and simple imputation like mean.

    b. 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.

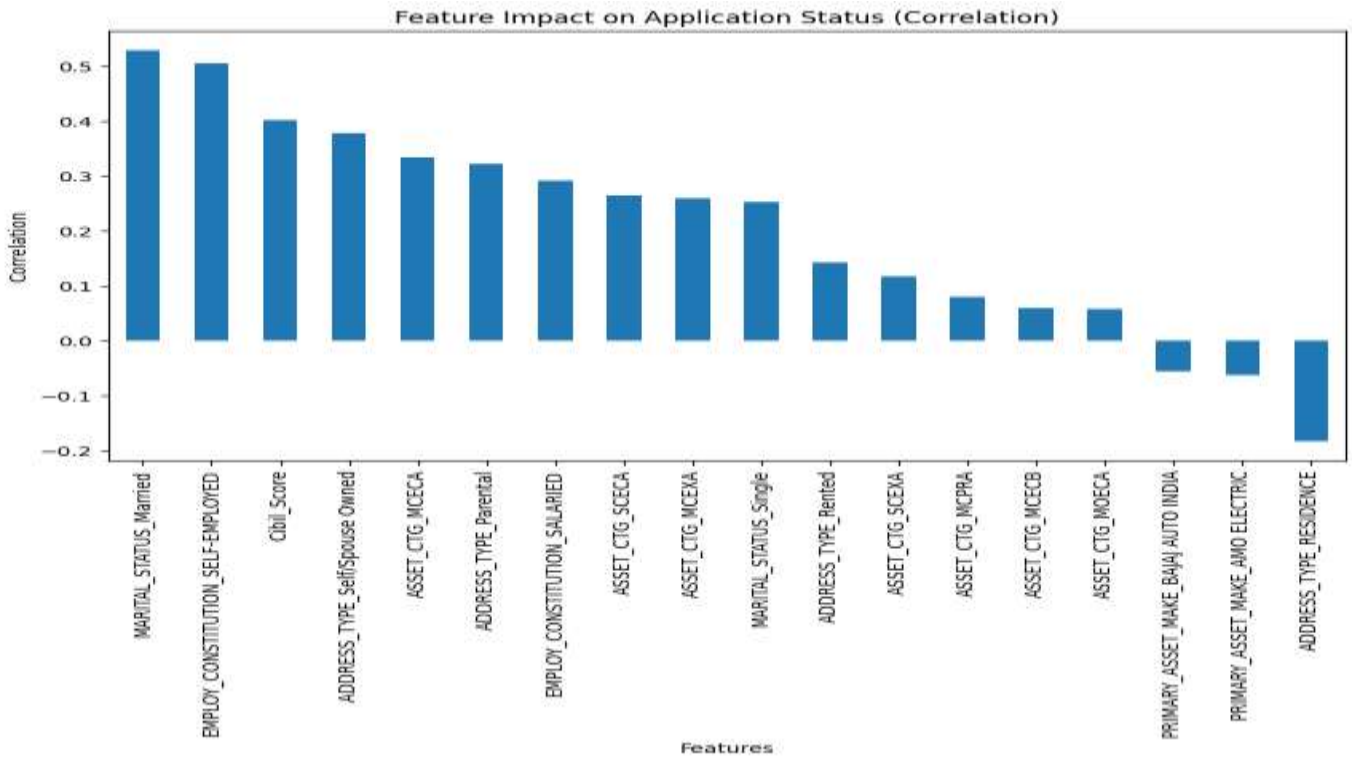
1. Important features like cibilScore asset categories, have a lot of missing data.
    a. Around 50% so it won't be wise to use simple imputation like mean or dropping out or any other statistical method.
    b. Will have to use advanced techniques like using the nearest neighbor value based on Euclidean distance.
2. A lot of irrelevant data in the data which adds extra dimensionality causing problems because of the curse of dimensionality.
    a. 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'`
    b. 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.
    a. `Cibil score, asset cost, phone digital age, phone name match score.`
4. Handling categorical columns.
    a. One hot encoding them.
5. Imputation.
    a. Simple imputation-: This gave an accuracy of over 94%, but it was not robust so dropping it.
    b. 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.
    a. One hot encoding of the data increased it to 127 columns
    b. Need to reduce it.
        i. Setting a threshold of 0.05 on the correlation between feature and target, and removing if it is less than that.

ii. PCA It gave the same result as the first method and it adds unexplainability and it is sensitive to outliers so dropped it.

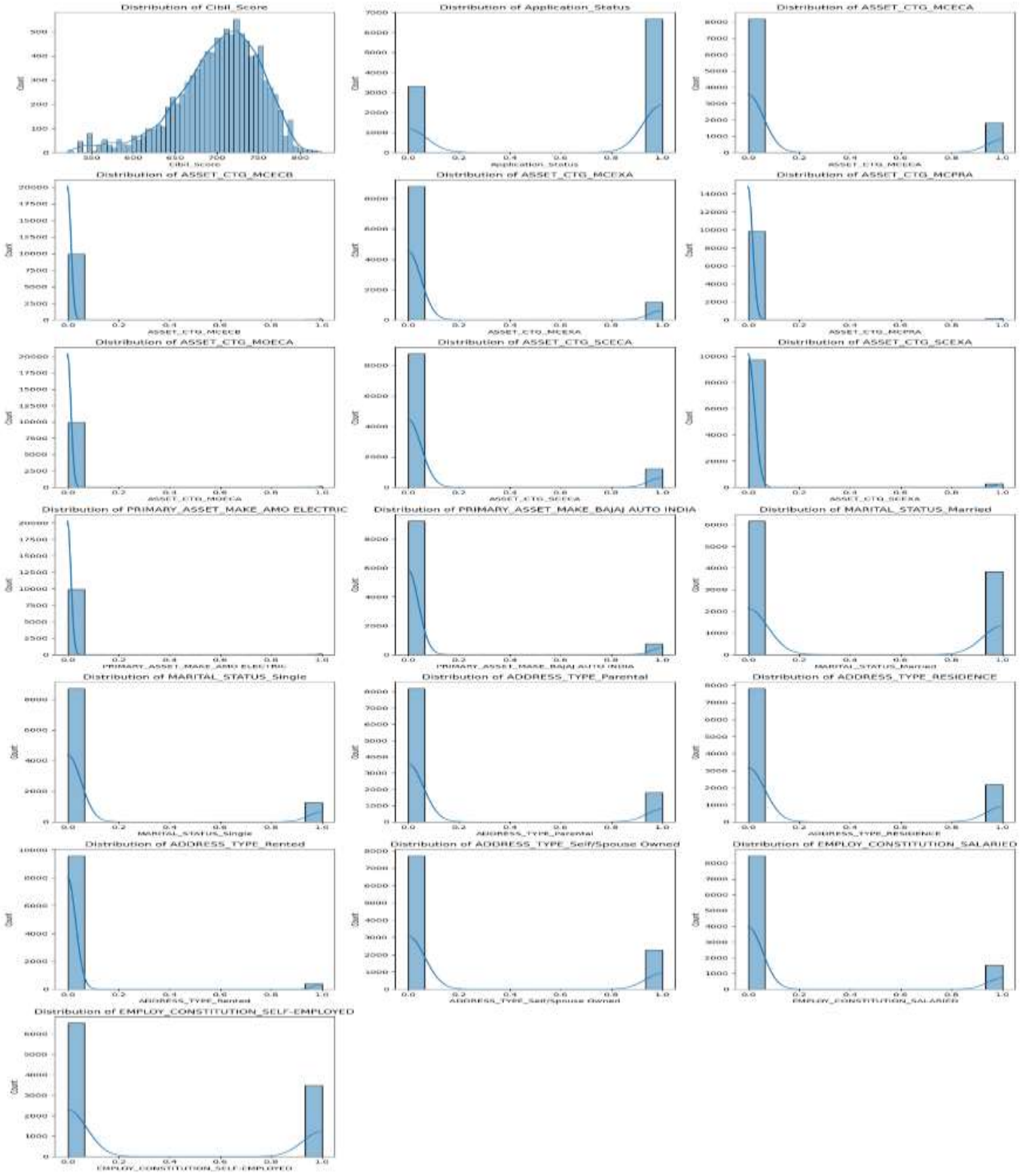
c. Reduced columns to 18 based on thresholding method.

EDA, insights and conclusion.

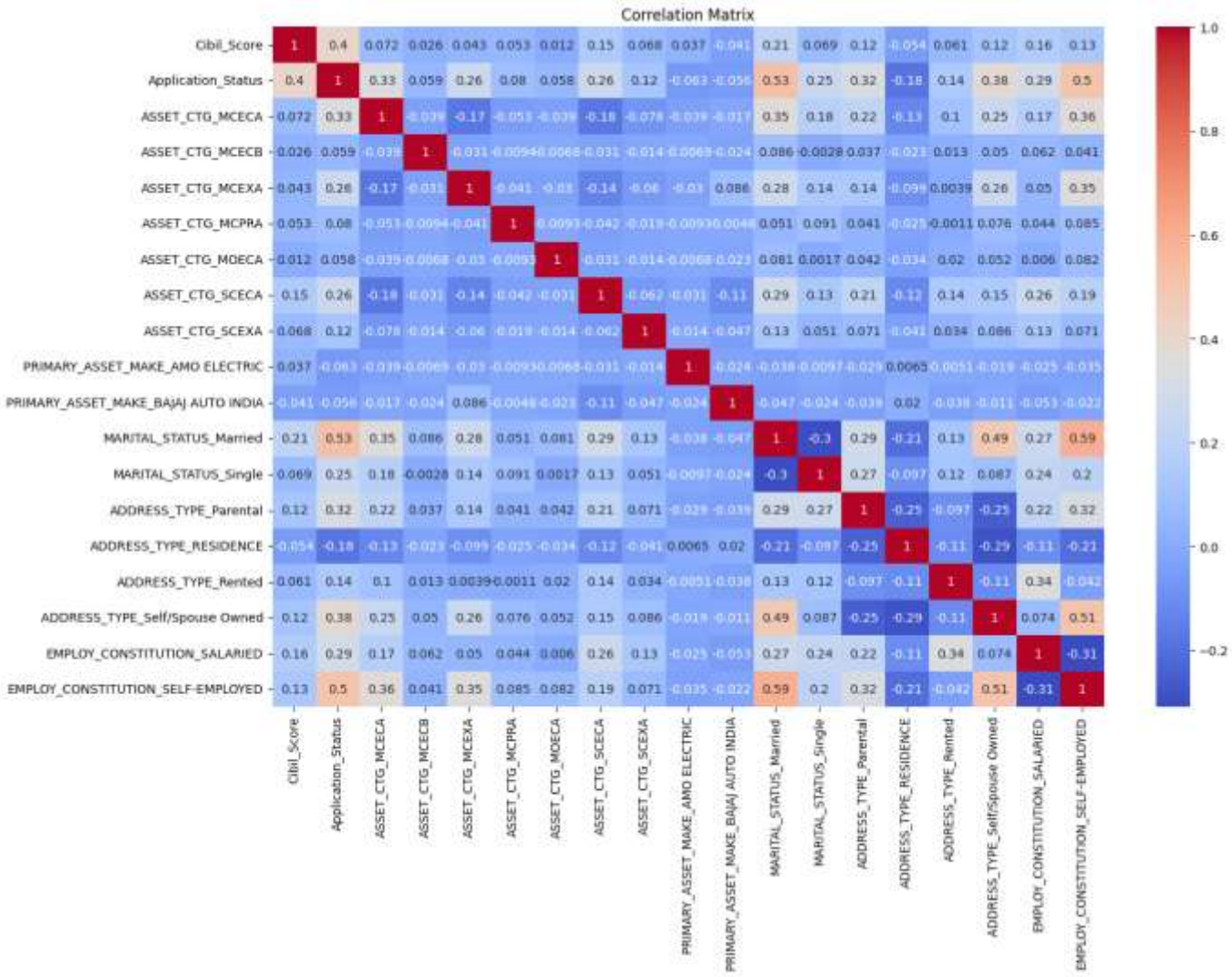
1. Correlation of features with application status {Check the notebook if not clear}.
    a. Being married, self employed, good cibil score, owning a house increases your chances loan aproval.



2. Imbalance in the data.

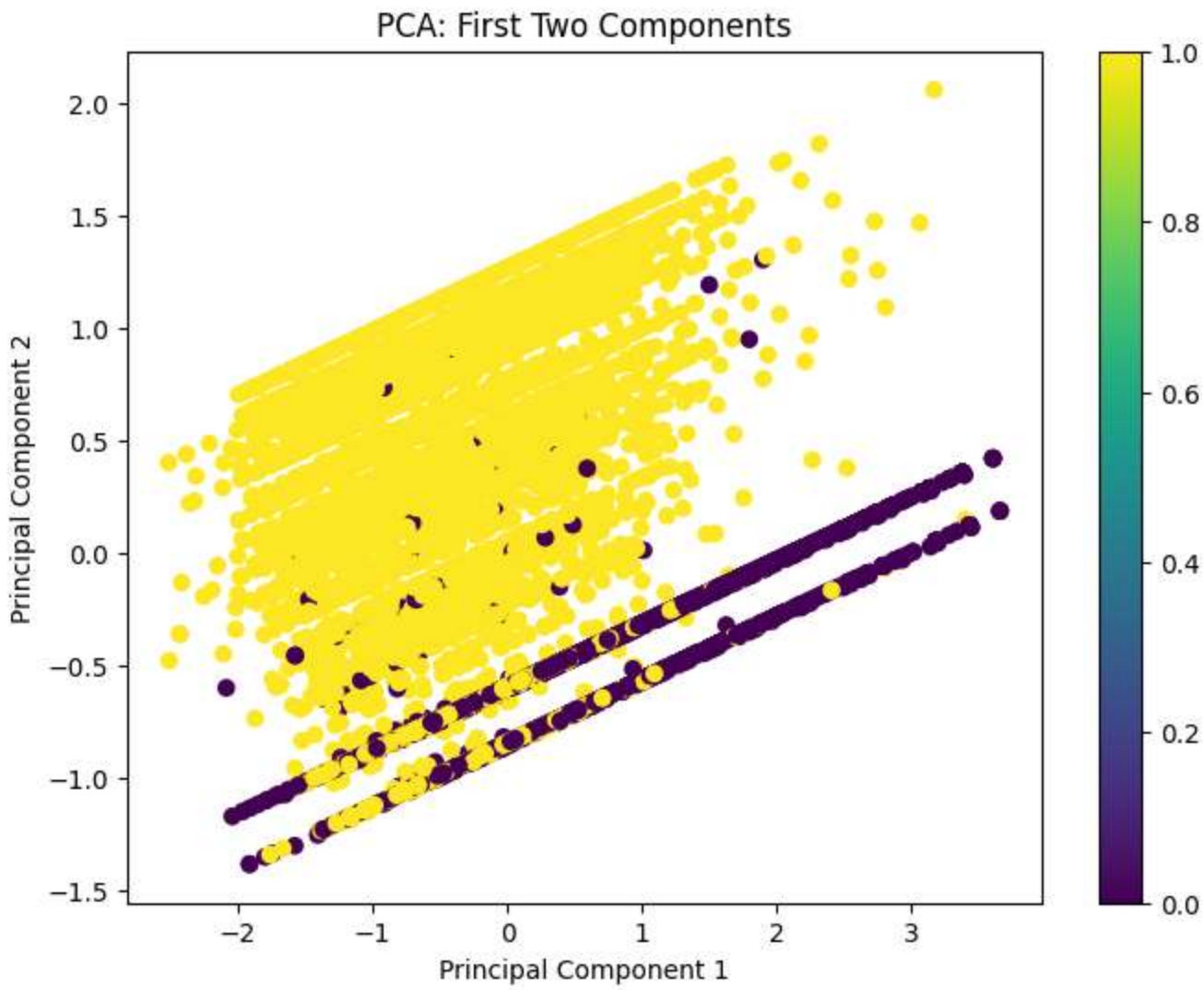


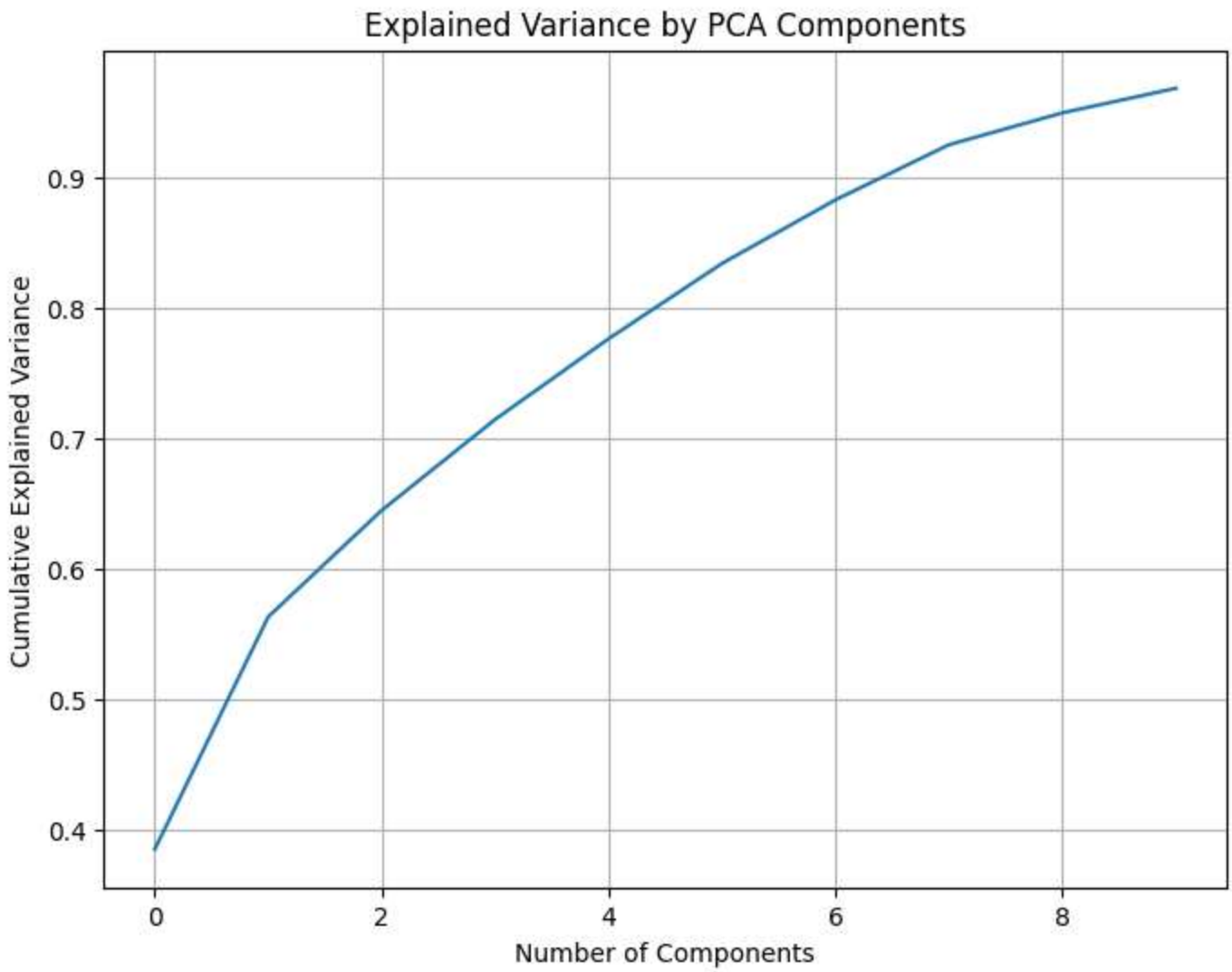
3. Correlation matrix between all the features, most of the diagonals are 1 and remaining are between -0.3 to +0.3 More improvements can be made..

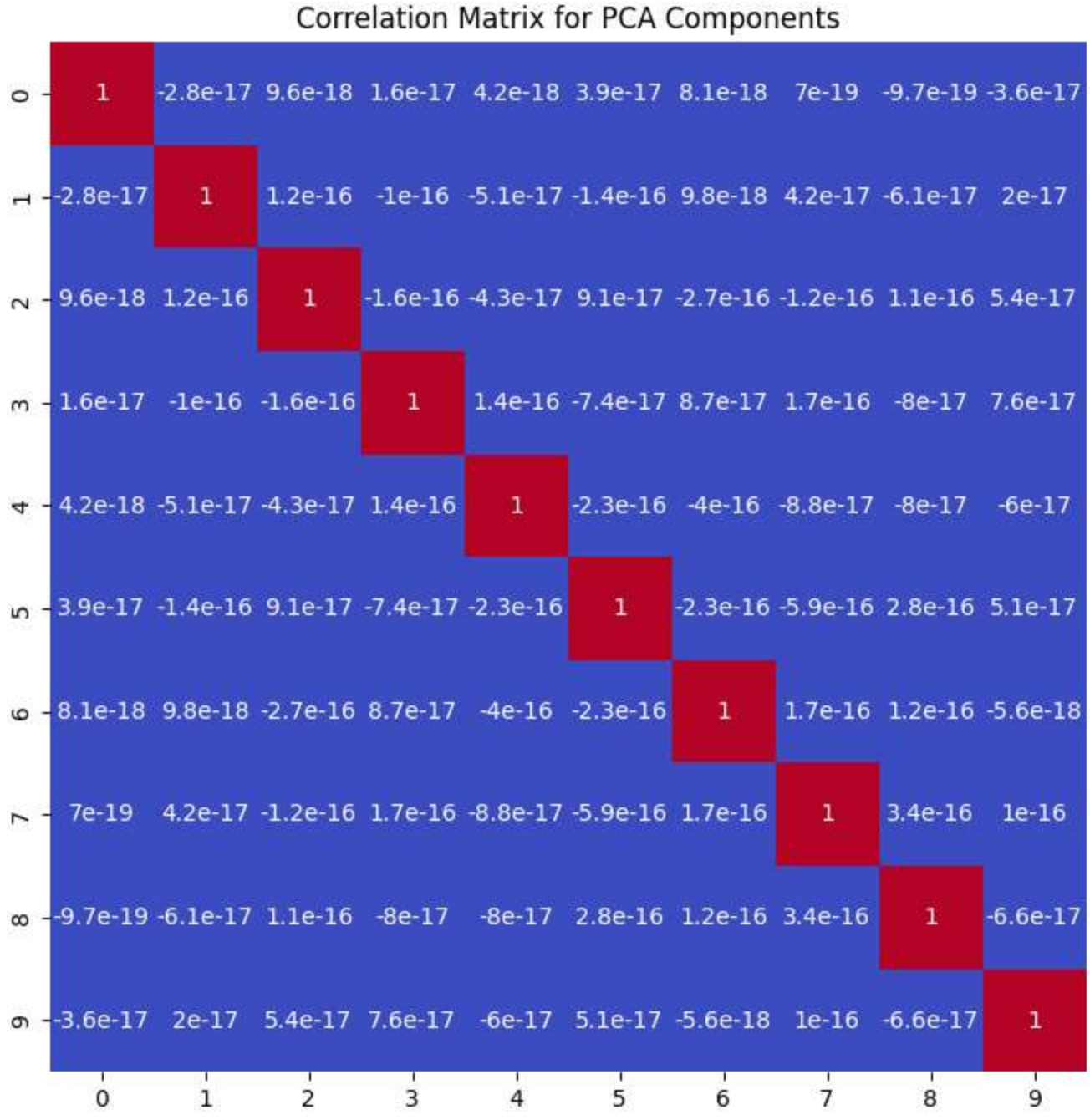


PCA and Dimensionality reduction.

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.


PCA: First Two Components
Principal Component 2
Principal Component 1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2
−1
0
1
2
3
1.0
0.8
0.6
0.4
0.2
0.0


Explained Variance by PCA Components
Cumulative Explained Variance
0.4
0.5
0.6
0.7
0.8
0.9
0
2
4
6
8
Number of Components

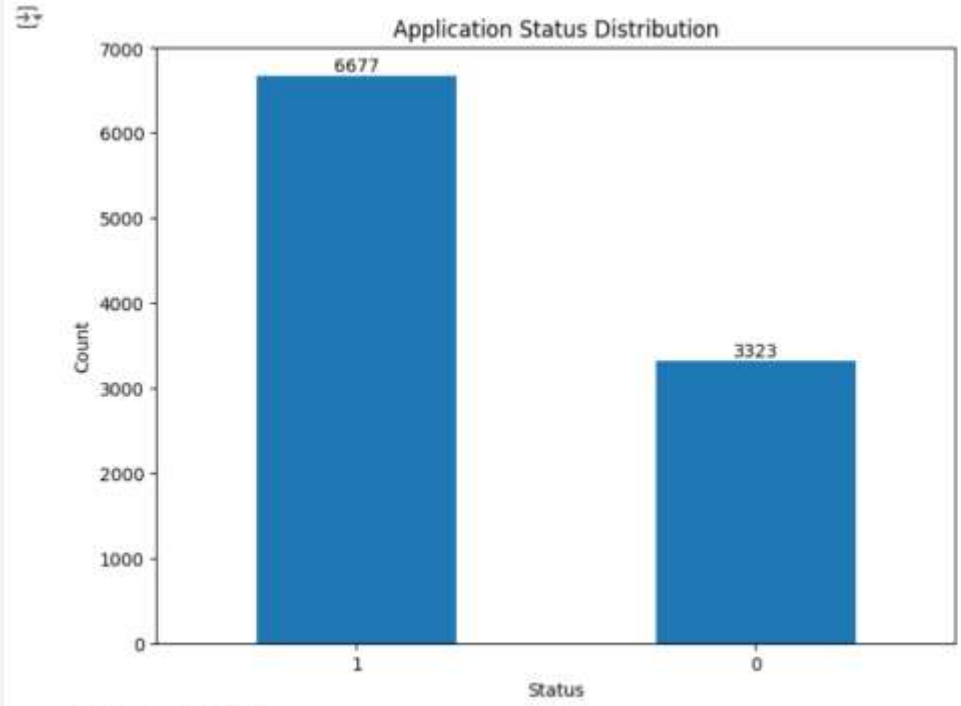

Correlation Matrix for PCA Components
0 1 -2.8e-17 9.6e-18 1.6e-17 4.2e-18 3.9e-17 8.1e-18 7e-19 -9.7e-19 -3.6e-17
1 -2.8e-17 1 1.2e-16 -1e-16 -5.1e-17 -1.4e-16 9.8e-18 4.2e-17 -6.1e-17 2e-17
2 9.6e-18 1.2e-16 1 -1.6e-16 -4.3e-17 9.1e-17 -2.7e-16 -1.2e-16 1.1e-16 5.4e-17
3 1.6e-17 -1e-16 -1.6e-16 1 1.4e-16 -7.4e-17 8.7e-17 1.7e-16 -8e-17 7.6e-17
4 4.2e-18 -5.1e-17 -4.3e-17 1.4e-16 1 -2.3e-16 -4e-16 -8.8e-17 -8e-17 -6e-17
5 3.9e-17 -1.4e-16 9.1e-17 -7.4e-17 -2.3e-16 1 -2.3e-16 -5.9e-16 2.8e-16 5.1e-17
6 8.1e-18 9.8e-18 -2.7e-16 8.7e-17 -4e-16 -2.3e-16 1 1.7e-16 1.2e-16 -5.6e-18
7 7e-19 4.2e-17 -1.2e-16 1.7e-16 -8.8e-17 -5.9e-16 1.7e-16 1 3.4e-16 1e-16
8 -9.7e-19 -6.1e-17 1.1e-16 -8e-17 -8e-17 2.8e-16 1.2e-16 3.4e-16 1 -6.6e-17
9 -3.6e-17 2e-17 5.4e-17 7.6e-17 -6e-17 5.1e-17 -5.6e-18 1e-16 -6.6e-17 1
0 1 2 3 4 5 6 7 8 9

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

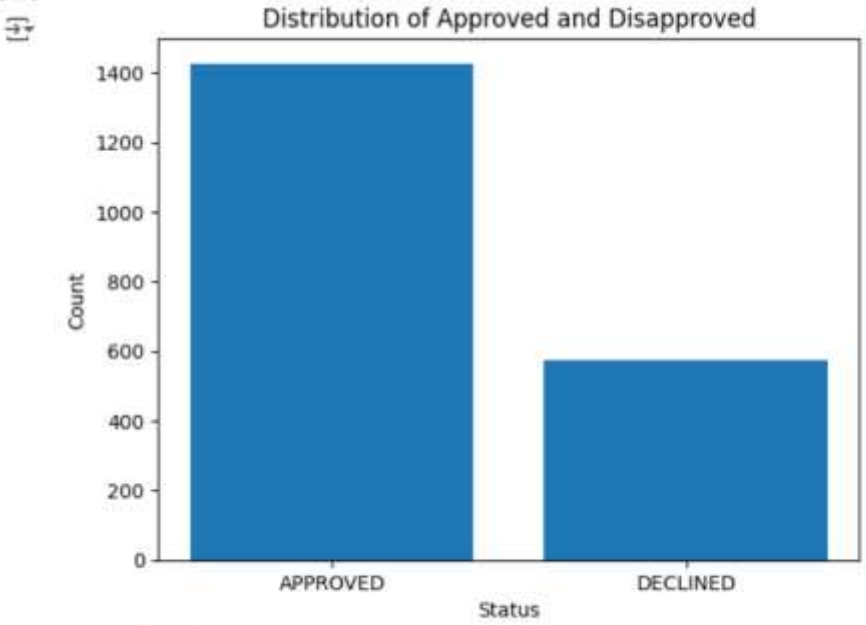
1. Training data.




```
Approved Ratio: 0.67
Disapproved Ratio: 0.33
```

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

```
[65]
```




```
Approved Ratio: 0.71
Disapproved Ratio: 0.29
```