CAPSTONE PROJECT - I

LOAN DEFAULT PREDICTION

PROBLEM STATEMENT

- Financial institutions, spanning from major banks to governmental entities, extensively utilize loan services. Among the foremost goals for entities offering financial loan services is the reduction of payment defaults and the assurance that borrowers adhere to the expected repayment terms. To achieve this, many organizations aim to pinpoint individuals with the greatest likelihood of defaulting on their loans.
- Limited visibility into potential risks associated with loans hampers the efficiency of decision-making processes for financial institutions. As we can't solely depend on the indicators like credit score to identify defaulters, we also need to consider other parameters/indicators.

AIM & OBJECTIVE :-

- This project aimed to develop a system for predicting loan default using both machine learning algorithms and data visualization tools, specifically Power BI. The combination of these will provide a better/proactive solution for financial institutions, enhancing decision-making processes and risk management.
- Also, these models and interactive dashboards will not only help predict and visualize the likelihood of loan default but also identify trends and patterns

METHODOLOGY :-

- Data Extraction & Cleaning:- Extract, transform, and load (ETL) relevant data into both Power BI for visualization and Python for preprocessing, and then into the machine learning model.
- Machine Learning Models:- Implement predictive models using algorithms such as logistic regression, decision trees, or ensemble methods to forecast loan default probabilities.
- Dashboard Design:- To identify key metrics/indicators and design visually compelling dashboards for analysis.

DATA PREPROCESSING:-

- The Purpose of preprocessing involves cleaning and transforming raw data into a format that can be effectively used for machine learning. The goal is to make the data suitable for the model by handling missing values, scaling features, encoding categorical variables, and addressing other issues.
- Some Key inferences after performing preprocessing:-
  - The dataset consists of 18 fields and 255,346 records, including integer and object data types. The 'objects' need to be converted into integers for further modeling.
  - The dataset is free of duplicate values and special characters.
  - However, it does contain null values, since the missing (null) values constitute less than 1%, we can drop them using dropna(). But it's important to note that some of these missing values may contribute to a default value of 1, indicating that these records show loan defaults. Therefore, we'll use fillna() to impute these missing values by employing the mode and median aggregation.
  - Next, we will utilize Label Encoder to convert categorical variables into numerical ones.
  - Following this, we'll split the data into independent (X) and dependent (y) variables. Subsequently, we'll scale the independent variables (X) using Standard Scaler.
  - Lastly, we will divide these variables into training and testing sets to facilitate further model building.

EDA :-

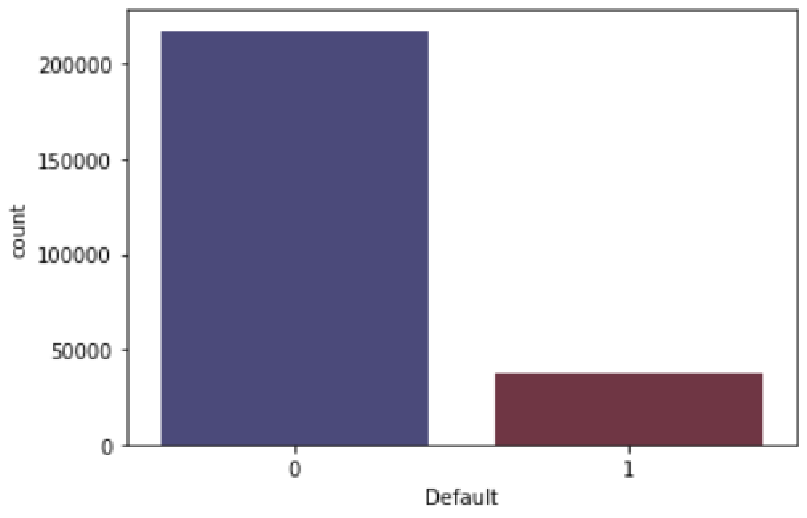
- EDA stands for Exploratory Data Analysis
- The Purpose of EDA is for analyzing and visualizing data to understand its key characteristics, patterns, and relationships. The goal is to gain insights into the data distribution, identify potential outliers, and understand the relationships between variables.
- EDA involves creating visualizations (histograms, scatter plots, etc.) to explore the distribution of individual features, relationships between features, and potential patterns in the data.
- EDA is typically performed after data preprocessing and before building machine learning models. It helps in understanding the data, selecting relevant features, and guiding subsequent modeling decisions.
- Mostly we'll deal with Univariate and Bivariate analysis.

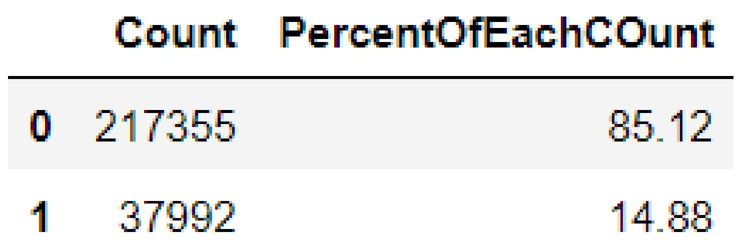
- <u>Univariate Analysis:-</u>

1. Distribution of target variable :-

   - ➢ The distribution of the default feature is visualized using a count plot

   - ➢ Additionally, I've generated a data frame that includes each unique value of the 'default' feature along with its corresponding percentage.





	Count	PercentOfEachCOunt
0	217355	85.12
1	37992	14.88

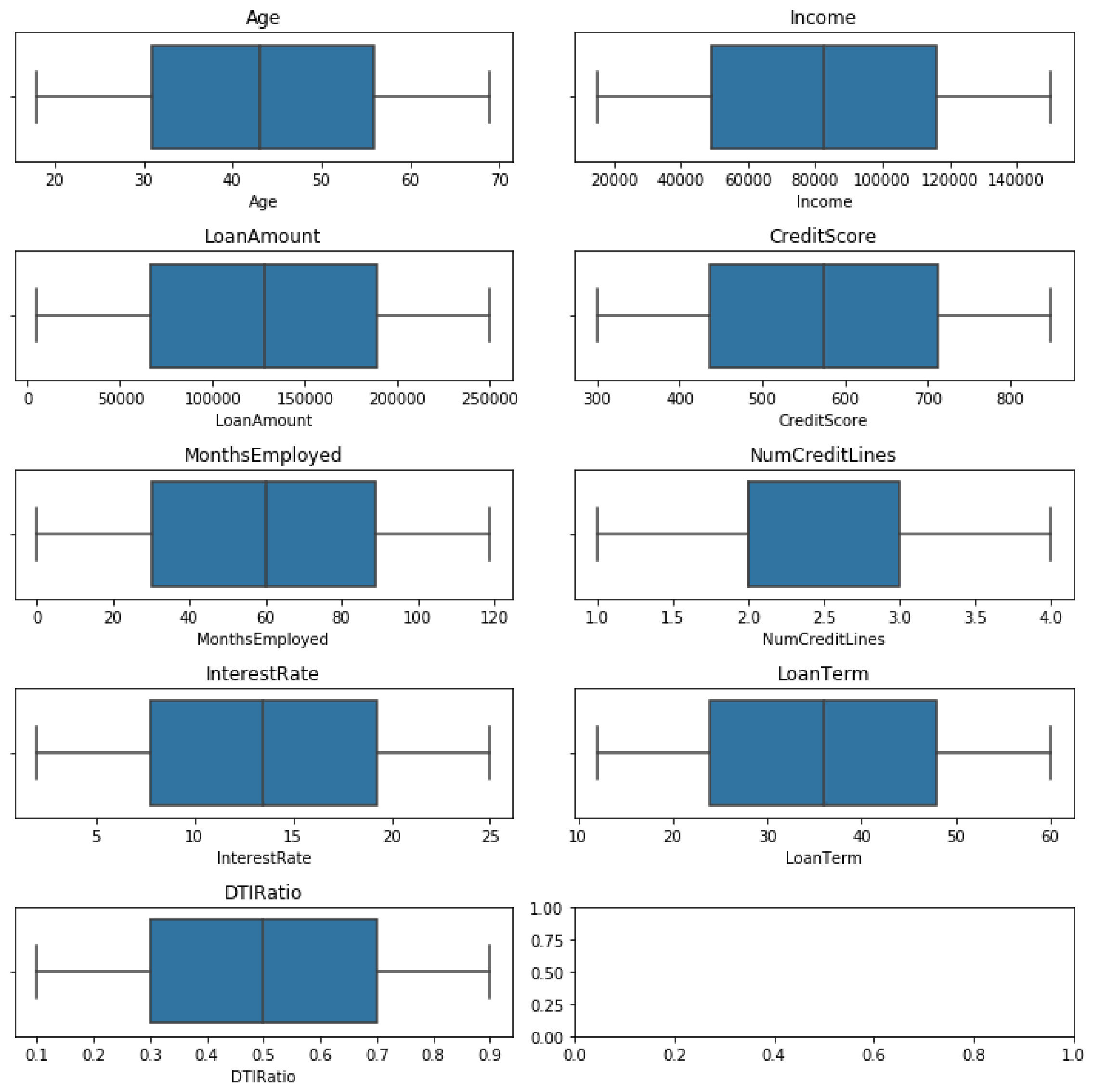
2 . Checking for Outliers:-

- Outliers are anomalies within a dataset, representing data points that deviate significantly from the majority. They can be either much larger or significantly smaller than other values

- Outliers may arise due to measurement errors, experimental variability, or genuine anomalies in the data.

- Identifying and handling outliers is crucial because they can profoundly impact the results of statistical analyses and machine learning models.

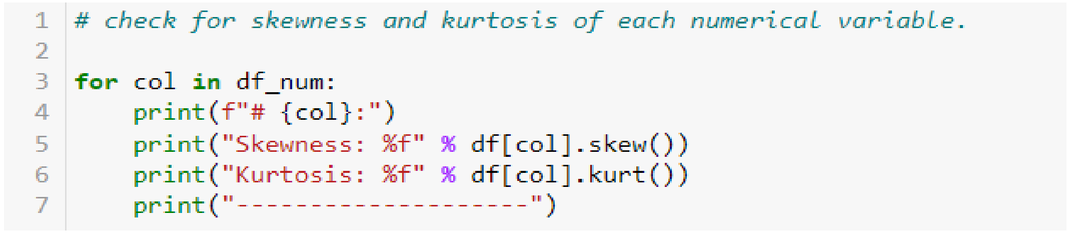
- Inference:-
  - No outliers were found in the dataset.



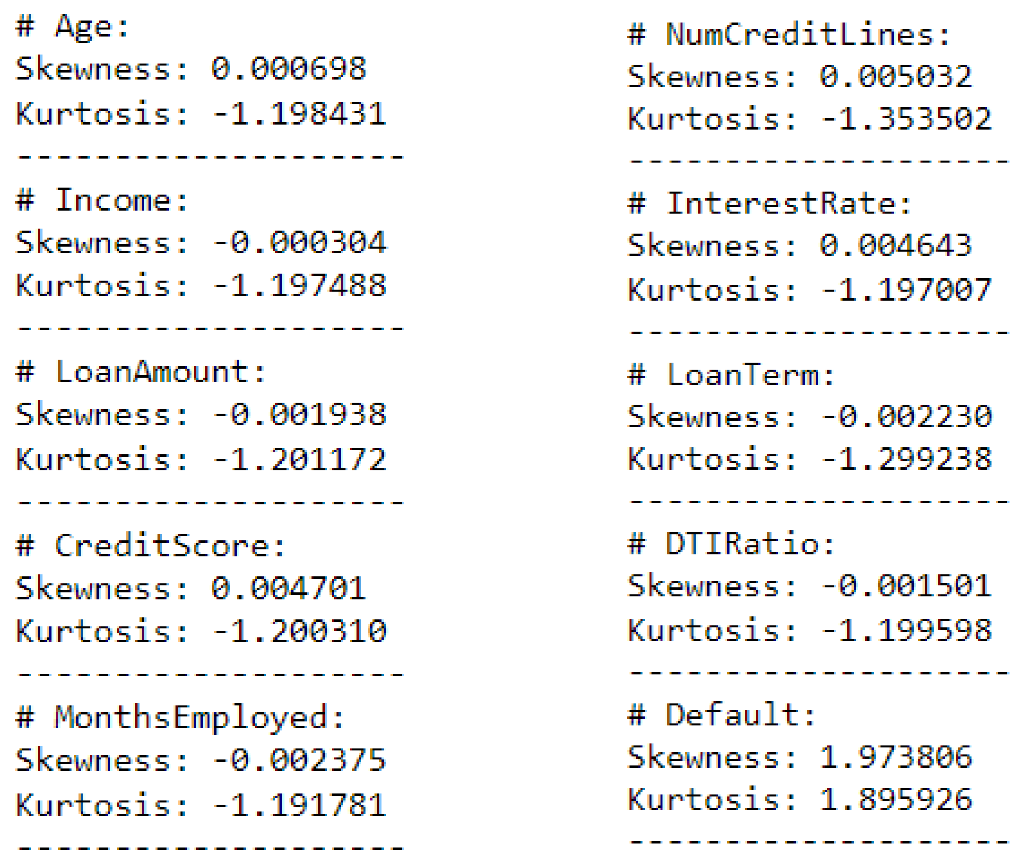
3. Skewness and Kurtosis:-

- Skewness measures the asymmetry of the data, indicating whether the distribution is skewed to the left (negatively skewed), to the right (positively skewed), or if it is symmetric
- Inference:-
  - Upon observation, it appears that all of the numerical features have skewness values near 0. This suggests that the distributions are nearly symmetrical.

- Kurtosis measures the tailedness or sharpness of a distribution's peak
- Inference:-
  - Upon observation, it appears that all of the numerical features have kurtosis values less than 0. This suggests that the distribution is platykurtic, characterized by lighter tails and a flatter peak.



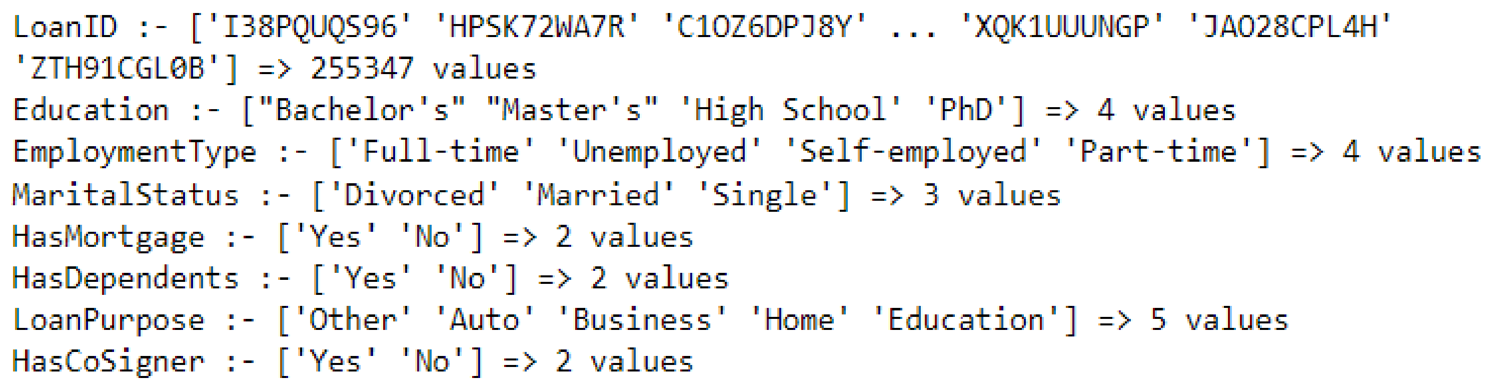
```python
# check for skewness and kurtosis of each numerical variable.

for col in df_num:
    print(f"# {col}:")
    print("Skewness: %f" % df[col].skew())
    print("Kurtosis: %f" % df[col].kurt())
    print("--------------------")
```



```
# Age:
Skewness: 0.000698
Kurtosis: -1.198431
--------------------
# Income:
Skewness: -0.000304
Kurtosis: -1.197488
--------------------
# LoanAmount:
Skewness: -0.001938
Kurtosis: -1.201172
--------------------
# CreditScore:
Skewness: 0.004701
Kurtosis: -1.200310
--------------------
# MonthsEmployed:
Skewness: -0.002375
Kurtosis: -1.191781
--------------------
```

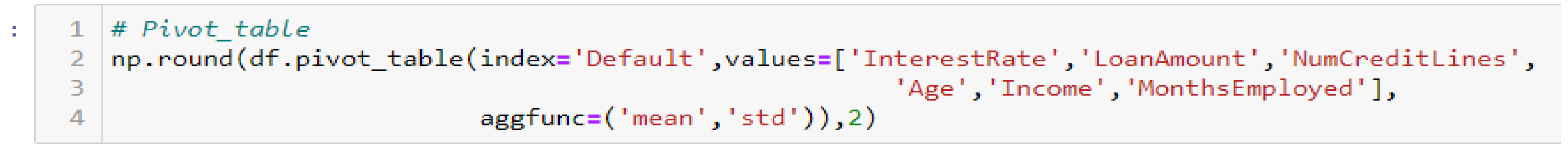
```
# NumCreditLines:
Skewness: 0.005032
Kurtosis: -1.353502
--------------------
# InterestRate:
Skewness: 0.004643
Kurtosis: -1.197007
--------------------
# LoanTerm:
Skewness: -0.002230
Kurtosis: -1.299238
--------------------
# DTIRatio:
Skewness: -0.001501
Kurtosis: -1.199598
--------------------
# Default:
Skewness: 1.973806
Kurtosis: 1.895926
--------------------
```

4 . Checking for unique values and its count in categorical variable:-

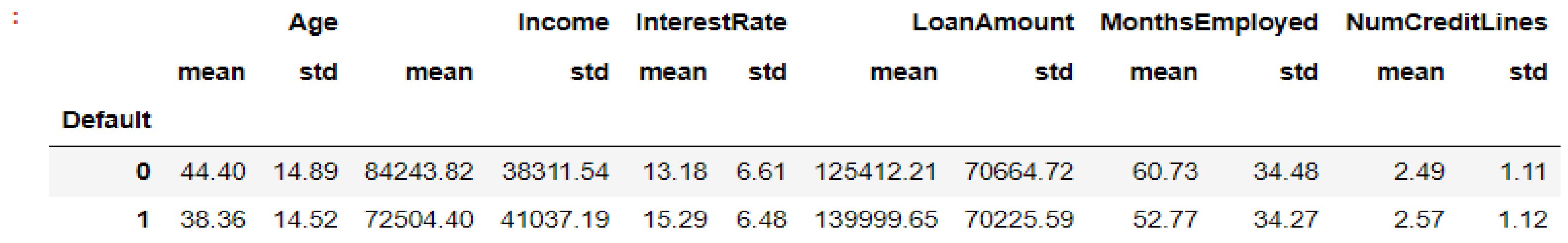


```
LoanID :- ['I38PQUQS96' 'HPSK72WA7R' 'C1OZ6DPJ8Y' ... 'XQK1UUUNGP' 'JAO28CPL4H'
'ZTH91CGL0B'] => 255347 values
Education :- ["Bachelor's" "Master's" 'High School' 'PhD'] => 4 values
EmploymentType :- ['Full-time' 'Unemployed' 'Self-employed' 'Part-time'] => 4 values
MaritalStatus :- ['Divorced' 'Married' 'Single'] => 3 values
HasMortgage :- ['Yes' 'No'] => 2 values
HasDependents :- ['Yes' 'No'] => 2 values
LoanPurpose :- ['Other' 'Auto' 'Business' 'Home' 'Education'] => 5 values
HasCoSigner :- ['Yes' 'No'] => 2 values
```

- Pivot table is used for reshaping and summarizing data, where you can define indices, columns, and values to aggregate over.

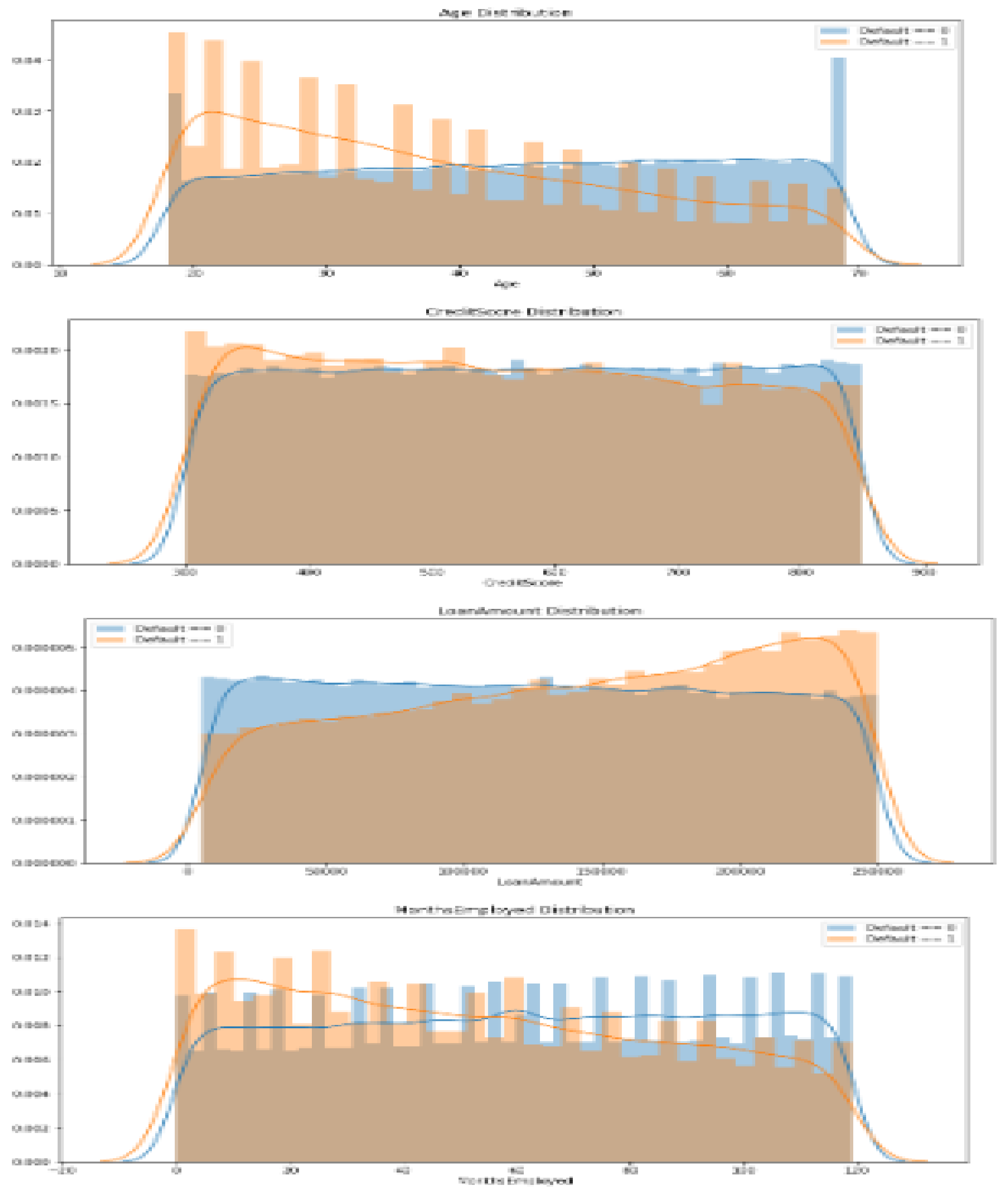


```python
# Pivot_table
np.round(df.pivot_table(index='Default',values=['InterestRate','LoanAmount','NumCreditLines',
                                                 'Age','Income','MonthsEmployed'],
                        aggfunc=('mean','std')),2)
```



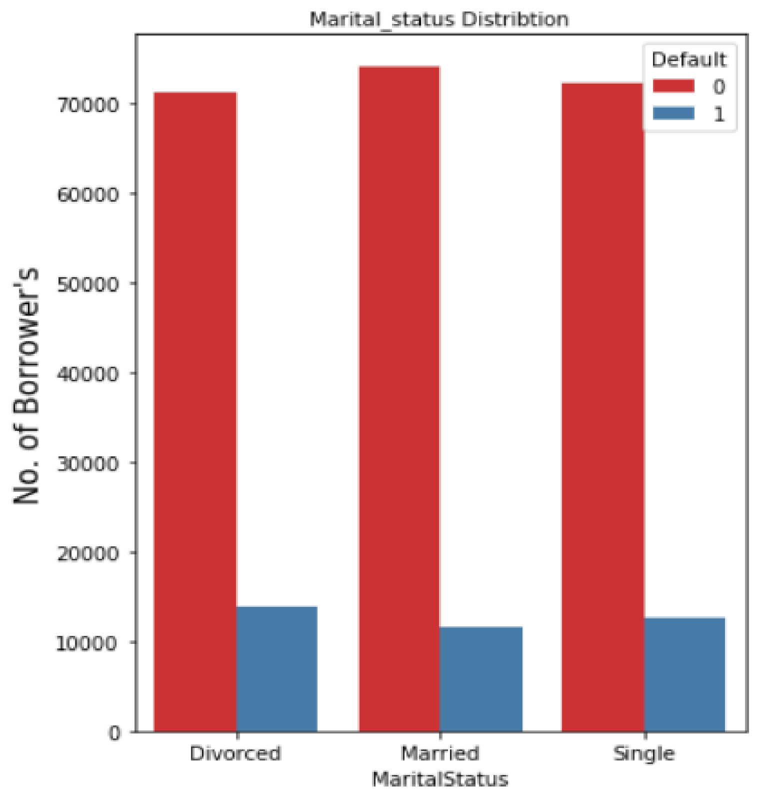
	Age		Income		InterestRate		LoanAmount		MonthsEmployed		NumCreditLines	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Default												
0	44.40	14.89	84243.82	38311.54	13.18	6.61	125412.21	70664.72	60.73	34.48	2.49	1.11
1	38.36	14.52	72504.40	41037.19	15.29	6.48	139999.65	70225.59	52.77	34.27	2.57	1.12

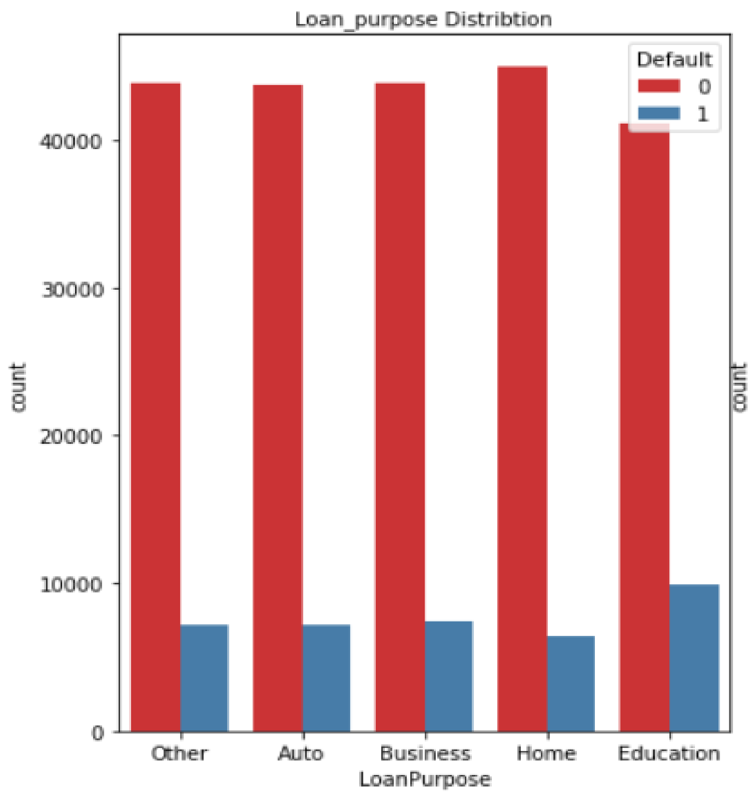
- <u>Bivariate Analysis:-</u>
  1. Distribution of Age, credit score, loan amount, Number of months employed using distribution plot.
     - ➢ Distribution plot combines a histogram with a Kernel Density Estimation (KDE) plot. When using distplot, the y-axis includes both the histogram (frequency or count) and the KDE (density).
     - ➢ Inferences:- From the graphs,
       - we can infer that the age group 20-24 has a high count, making individuals in this age range more likely to default compared to those in other income bands.
       - Additionally, we observe that individuals with a credit score below 400 are more likely to default.
       - Moreover, borrowers with a loan amount higher than $150,000 face difficulties in paying off the loan.
       - Furthermore, individuals with 0 to 18 months of employment tend to default more frequently

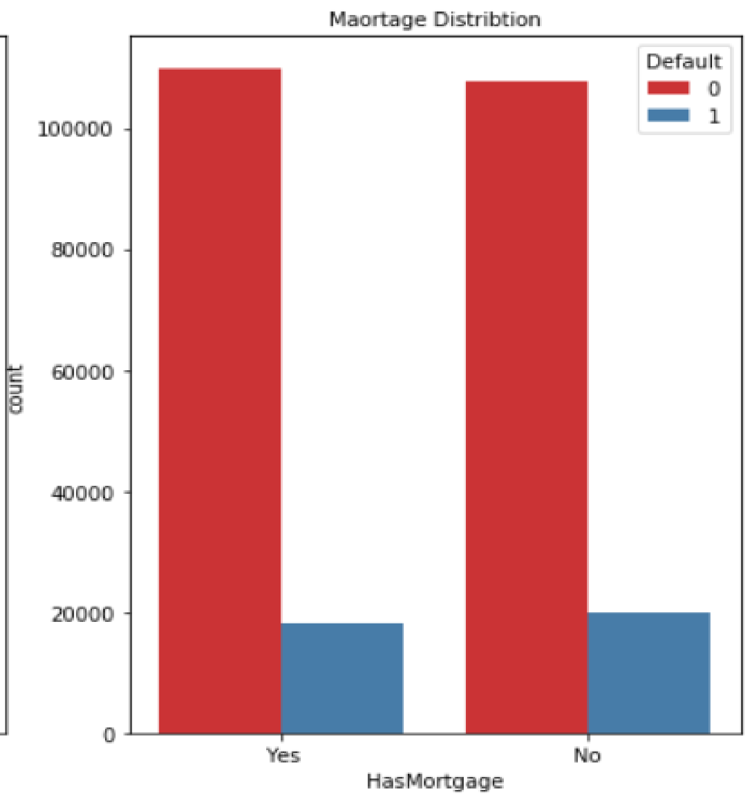


2. Distribution of Marital status, loan purpose, mortgage by default using count plot

- Count plot is used to visualize the frequency distribution of categorical variable.
- Inferences:-
  - We can see that the chance of defaulting for married individuals is relatively low compared to those who are divorced or single. Additionally, the default rate for loans taken for educational purposes is higher compared to other purposes. Furthermore, borrowers with or without a mortgage have an almost equal chance of defaulting

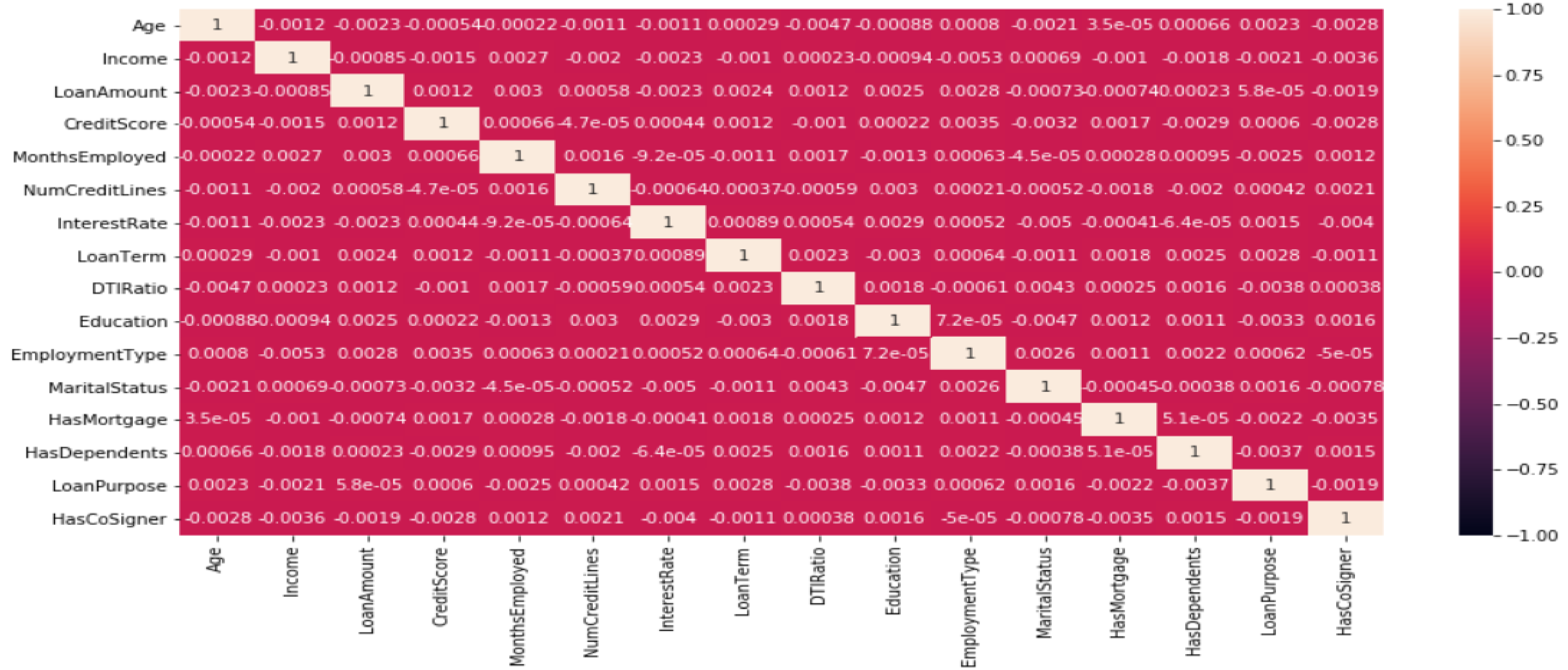






3 . Checking for Multicollinearity between independent (predictor) variables using heat map

- Multicollinearity is a statistical concept that occurs when two or more independent variables in a model are highly correlated with each other. It indicates a strong linear relationship among the predictor variables
- Heatmaps are visual representations of data in a matrix format, where colors represent values. In the context of multicollinearity, heatmaps are often used to display the correlation matrix of independent variables.



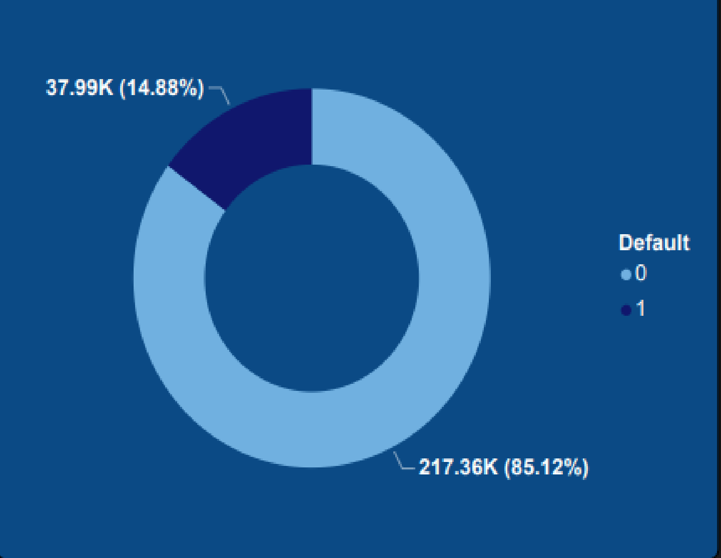
	Age	Income	LoanAmount	CreditScore	MonthsEmployed	NumCreditLines	InterestRate	LoanTerm	DTIRatio	Education	EmploymentType	MaritalStatus	HasMortgage	HasDependents	LoanPurpose	HasCoSigner
Age	1	-0.0012	-0.0023	-0.00054	-0.00022	-0.0011	-0.0011	0.00029	-0.0047	-0.00088	0.0008	-0.0021	3.5e-05	0.00066	0.0023	-0.0028
Income	-0.0012	1	-0.00085	-0.0015	0.0027	-0.002	-0.0023	-0.001	0.00023	-0.00094	-0.0053	0.00069	-0.001	-0.0018	-0.0021	-0.0036
LoanAmount	-0.0023	-0.00085	1	0.0012	0.003	0.00058	-0.0023	0.0024	0.0012	0.0025	0.0028	-0.00073	-0.00074	0.00023	5.8e-05	-0.0019
CreditScore	-0.00054	-0.0015	0.0012	1	0.00066	-4.7e-05	0.00044	0.0012	-0.001	0.00022	0.0035	-0.0032	0.0017	-0.0029	0.0006	-0.0028
MonthsEmployed	-0.00022	0.0027	0.003	0.00066	1	0.0016	-9.2e-05	-0.0011	0.0017	-0.0013	0.00063	-4.5e-05	0.00028	0.00095	-0.0025	0.0012
NumCreditLines	-0.0011	-0.002	0.00058	-4.7e-05	0.0016	1	-0.00064	-0.00037	-0.00059	0.003	0.00021	-0.00052	-0.0018	-0.002	0.00042	0.0021
InterestRate	-0.0011	-0.0023	-0.0023	0.00044	-9.2e-05	-0.00064	1	0.00089	0.00054	0.0029	0.00052	-0.005	-0.00041	-6.4e-05	0.0015	-0.004
LoanTerm	0.00029	-0.001	0.0024	0.0012	-0.0011	-0.00037	0.00089	1	0.0023	-0.003	0.00064	-0.0011	0.0018	0.0025	0.0028	-0.0011
DTIRatio	-0.0047	0.00023	0.0012	-0.001	0.0017	-0.00059	0.00054	0.0023	1	0.0018	-0.00061	0.0043	0.00025	0.0016	-0.0038	0.00038
Education	-0.00088	-0.00094	0.0025	0.00022	-0.0013	0.003	0.0029	-0.003	0.0018	1	7.2e-05	-0.0047	0.0012	0.0011	-0.0033	0.0016
EmploymentType	0.0008	-0.0053	0.0028	0.0035	0.00063	0.00021	0.00052	0.00064	-0.00061	7.2e-05	1	0.0026	0.0011	0.0022	0.00062	-5e-05
MaritalStatus	-0.0021	0.00069	-0.00073	-0.0032	-4.5e-05	-0.00052	-0.005	-0.0011	0.0043	-0.0047	0.0026	1	-0.00045	-0.00038	0.0016	-0.00078
HasMortgage	3.5e-05	-0.001	-0.00074	0.0017	0.00028	-0.0018	-0.00041	0.0018	0.00025	0.0012	0.0011	-0.00045	1	5.1e-05	-0.0022	-0.0035
HasDependents	0.00066	-0.0018	0.00023	-0.0029	0.00095	-0.002	-6.4e-05	0.0025	0.0016	0.0011	0.0022	-0.00038	5.1e-05	1	-0.0037	0.0015
LoanPurpose	0.0023	-0.0021	5.8e-05	0.0006	-0.0025	0.00042	0.0015	0.0028	-0.0038	-0.0033	0.00062	0.0016	-0.0022	-0.0037	1	-0.0019
HasCoSigner	-0.0028	-0.0036	-0.0019	-0.0028	0.0012	0.0021	-0.004	-0.0011	0.00038	0.0016	-5e-05	-0.00078	-0.0035	0.0015	-0.0019	1

POWER BI & DASHBOARD

- Power BI is a business analytics tool developed by Microsoft that enables users to visualize and analyze data, share insights across an organization. It provides a suite of tools for data transformation, modeling, visualization, and sharing.
- A dashboard in Power BI is a collection of visualizations and reports organized on a single page, providing a high-level overview of key metrics and indicators for enhancing the decision-making process.

Loan default Analysis

Distribution of Defaulters



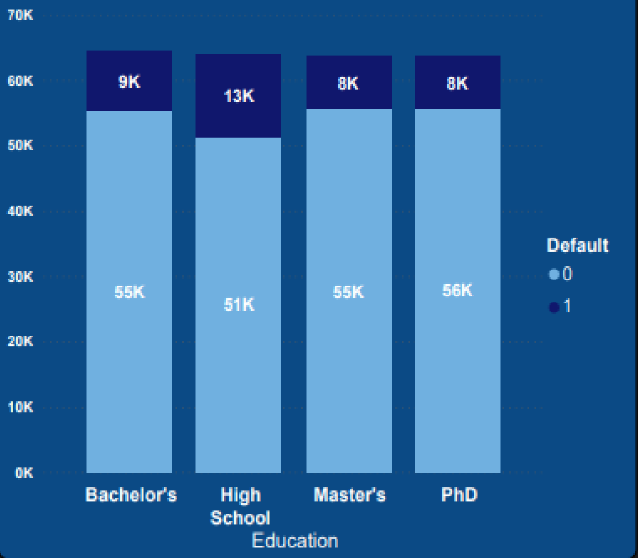
255K
Borrower's Count

574.26
Average of CreditScore

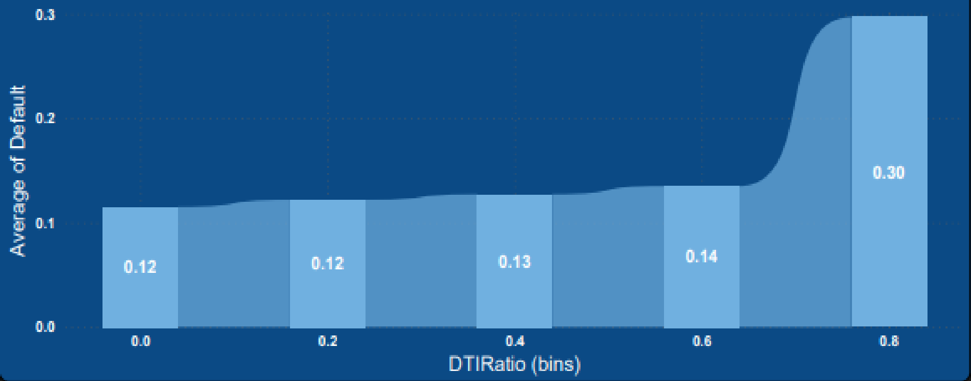
82K
Average of Income

0.50
Average of DTIRatio

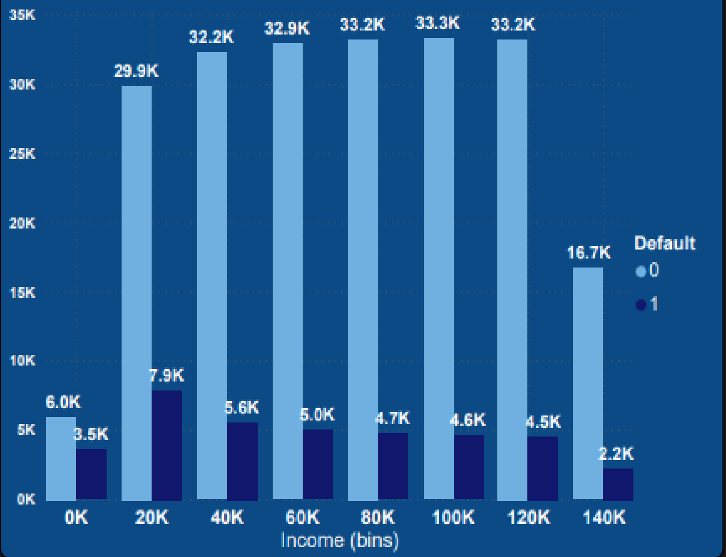
Distribution of Loan by Education & Default



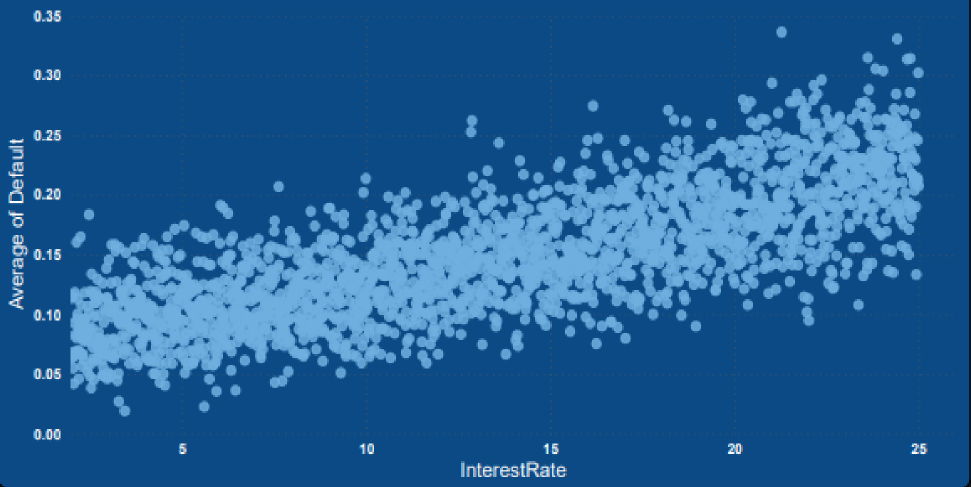
Average of Default by DTIRatio (bins)



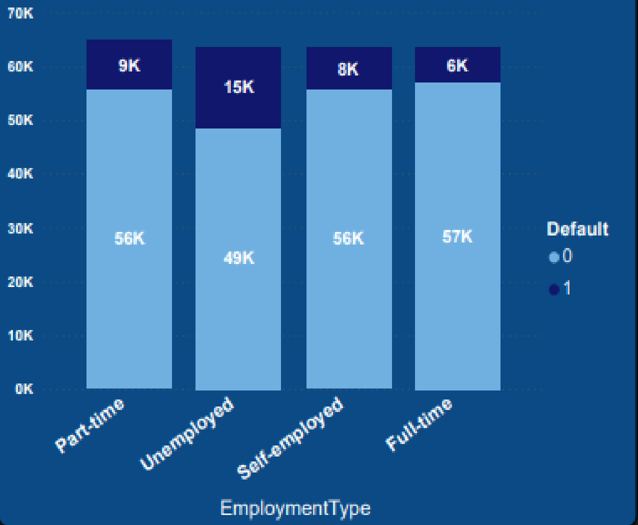
Income Distribution by Defaults



Average of Default by InterestRate



Distribution of Loan by EmploymentType and Default

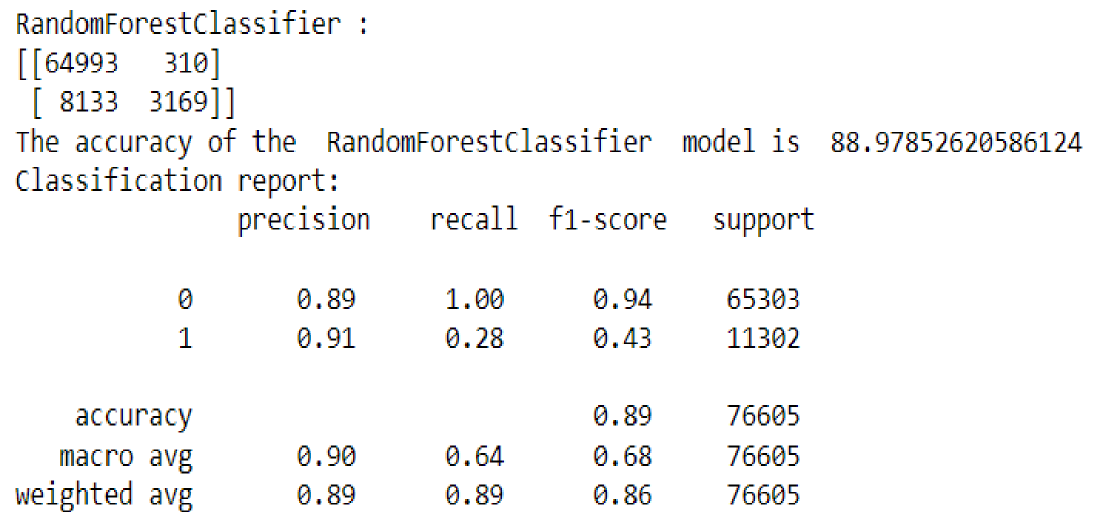


MODEL BUILDING:-

- Once all preprocessing and visualization are done, we can move ahead with building our predictive model.
- I have employed an 'all algorithms' approach, where I will be applying different algorithms to determine which one achieves the highest accuracy for modeling and, if possible, for tuning.
- The classification algorithms used include Decision Tree, K-Neighbors, Random Forest, and Logistic Regression.

❑ Inferences :

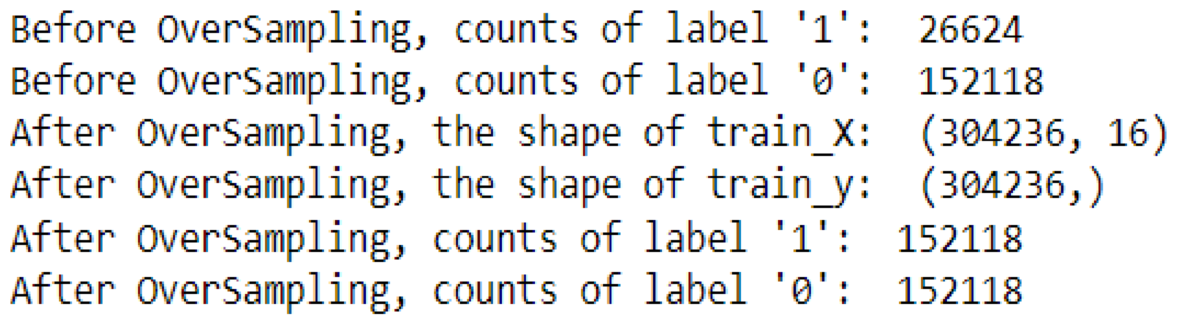
- Among the models used, Random Forest Classifier has achieved the highest accuracy, with achieving 88.98%.
- Additionally, we can infer that the lower accuracy of these models may be attributed to the low recall for class 1, possibly due to the relatively less amount of data in class 1 when compared to class 2.
- Hence, we can use SMOTE to address the imbalance in the data and evaluate whether it leads to improvements in recall and accuracy.



```
RandomForestClassifier :
[[64993   310]
 [ 8133  3169]]
The accuracy of the  RandomForestClassifier  model is  88.97852620586124
Classification report:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94     65303
           1       0.91      0.28      0.43     11302

    accuracy                           0.89     76605
   macro avg       0.90      0.64      0.68     76605
weighted avg       0.89      0.89      0.86     76605
```

HANDLING IMBALANCE DATA

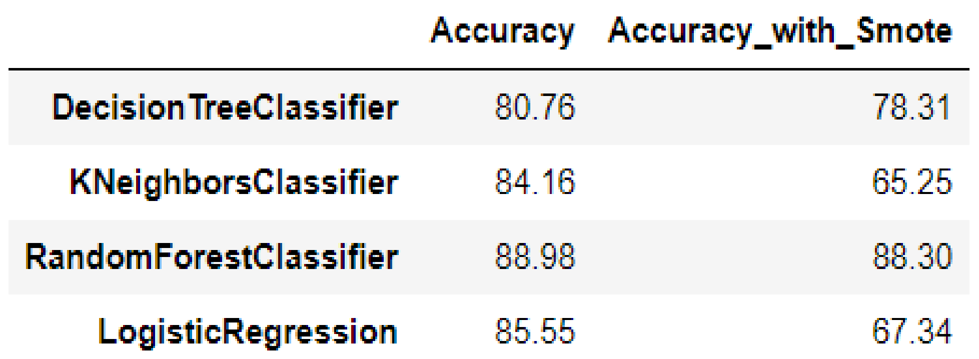
- SMOTE stands for Synthetic minority oversampling Technique. This technique allows us to up sample the minority class observation to reach the level of majority class by creating synthetic samples similar to existing samples
- The synthetic samples are created by taking the average values of k randomly selected data points.



```
Before OverSampling, counts of label '1':  26624
Before OverSampling, counts of label '0':  152118
After OverSampling, the shape of train_X:  (304236, 16)
After OverSampling, the shape of train_y:  (304236,)
After OverSampling, counts of label '1':  152118
After OverSampling, counts of label '0':  152118
```

❑ Inferences:-

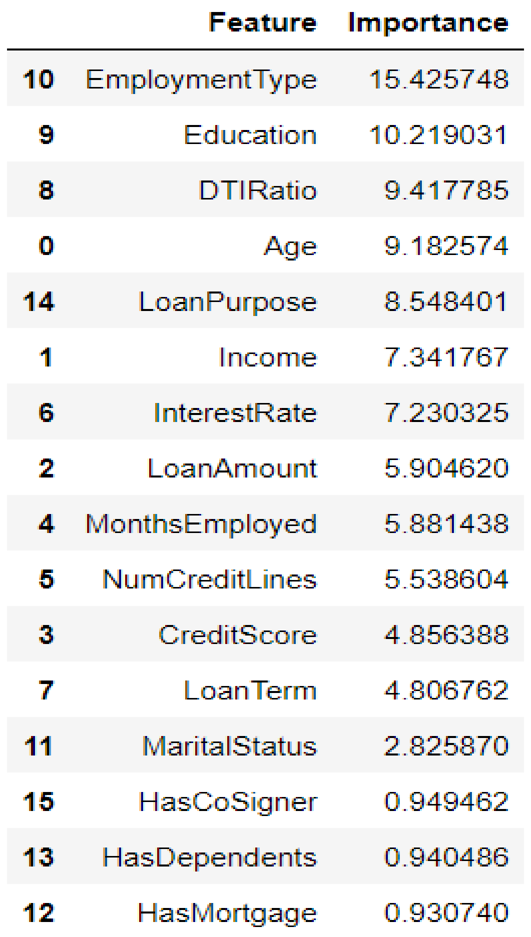
- In general, applying the SMOTE optimization technique has resulted in a slight increase in the recall of the minority class. However, the overall model accuracy shows a more noticeable decrease in comparison to the gain in recall. Notably, the accuracy of the best model, Random Forest, has only marginally decreased, approximately 0.7%, while the recall of class 1 has increased by about 0.02.



	Accuracy	Accuracy_with_Smote
DecisionTreeClassifier	80.76	78.31
KNeighborsClassifier	84.16	65.25
RandomForestClassifier	88.98	88.30
LogisticRegression	85.55	67.34

TUNING BASE MODEL

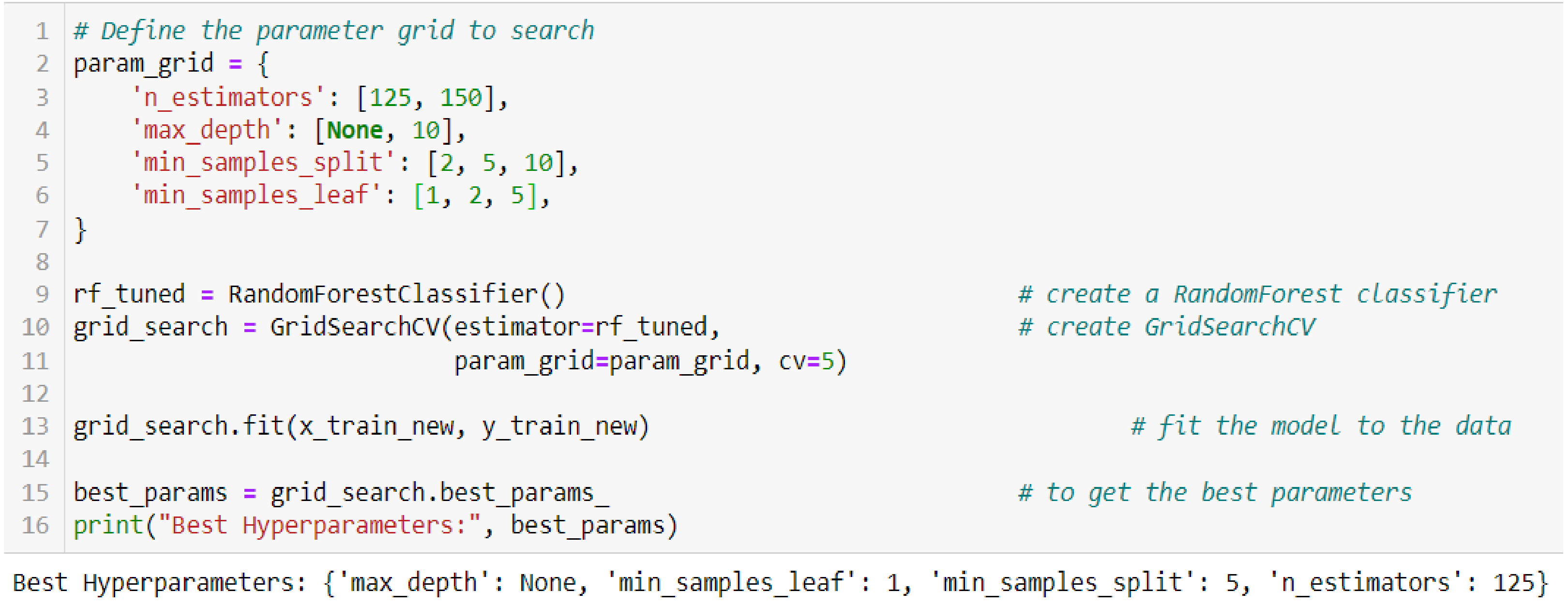
- Before finalizing the model, we will proceed to the second stage of feature selection and attempt to fine-tune the best model to explore its potential for further improvement.
- For selecting the best features, I have used an inbuilt parameter of random forest, 'feature_importances_'. This allows us to determine the percentage of parameters contributing to model building.
- Hence, using the best parameters, I have rebuilt the model, where the accuracy and recall are 89.05% and 0.28, respectively. This result is by far the best achieved.
- Further, I also plan to tune the hyperparameters of the Random Forest model using the Grid SearchCV technique, on these rebuild model not incorporating the SMOTE samples.



	Feature	Importance
10	EmploymentType	15.425748
9	Education	10.219031
8	DTIRatio	9.417785
0	Age	9.182574
14	LoanPurpose	8.548401
1	Income	7.341767
6	InterestRate	7.230325
2	LoanAmount	5.904620
4	MonthsEmployed	5.881438
5	NumCreditLines	5.538604
3	CreditScore	4.856388
7	LoanTerm	4.806762
11	MaritalStatus	2.825870
15	HasCoSigner	0.949462
13	HasDependents	0.940486
12	HasMortgage	0.930740

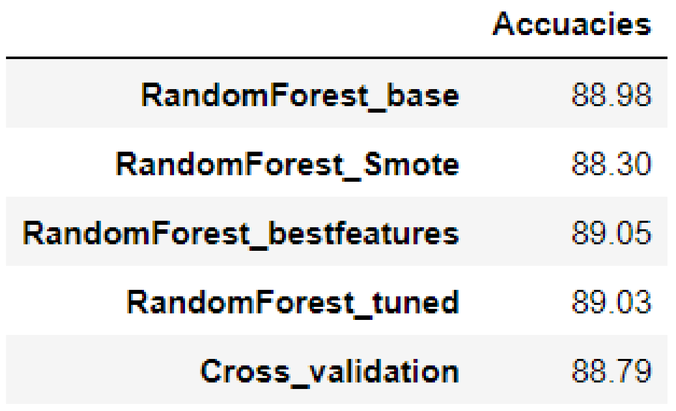
Hyperparameter Tuning using GridSearchCV

- GridSearchCV is a hyperparameter tuning technique in machine learning used to systematically search and select the best combination of hyperparameters for a model.
- The best options for some hyperparameters have been obtained, and using those, the model will be rebuilt.
- Consequently, the accuracy for the tuned model is found to be 89.03% with a recall of 0.28, which is slightly lower than our base model.
- Out of the models evaluated, the <u>Random Forest Classifier without SMOTE, optimized with only the top 13 important features</u>, performed the best. It achieved an accuracy of 89.05% with a recall value of 0.28. As a result, we have selected this model to predict on future unseen data.



```
# Define the parameter grid to search
param_grid = {
    'n_estimators': [125, 150],
    'max_depth': [None, 10],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 5],
}

rf_tuned = RandomForestClassifier()                                   # create a RandomForest classifier
grid_search = GridSearchCV(estimator=rf_tuned,                        # create GridSearchCV
                           param_grid=param_grid, cv=5)

grid_search.fit(x_train_new, y_train_new)                                 # fit the model to the data

best_params = grid_search.best_params_                                # to get the best parameters
print("Best Hyperparameters:", best_params)
```

```
Best Hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 125}
```



	Accuacies
RandomForest_base	88.98
RandomForest_Smote	88.30
RandomForest_bestfeatures	89.05
RandomForest_tuned	89.03
Cross_validation	88.79