CREDITFLOW AI: RISK-AGNOSTIC LENDING PLATFORM

Name: Arya Jain

Batch: 01 June Batch

Duration: 6 Months

Course: Data Science/AIML

CONTENT TABLE

3.4. DATA VISUALIZATION, STORYTELLING & EXPERIMENTING WITH CHARTS: UNDERSTAND THE RELATIONSHIPS BETWEEN VARIABLES

3.4.1. CHART - 1 & 2 (TARGET VARIABLE ANALYSIS)

3.4.2. CHART - 3, 4 & 5 (KEY FINANCIAL DISTRIBUTION ANALYSIS)

3.4.3. CHART - 6 & 7 (CATEGORICAL FEATURE DISTRIBUTIONS)

3.4.4. CHART - 8 & 9 (CATEGORICAL IMPACT ON TARGETS)

3.4.5. CHART - 10, 11, 12 & 13 (NUMERICAL RELATIONSHIPS AND DISTRIBUTIONS)

3.4.6. CHART - 14 - CORRELATION HEATMAP

3.4.7. CHART - 15 - PAIR PLOT

3.5. HYPOTHESIS TESTING

3.5.1. HYPOTHETICAL STATEMENT - 1

3.5.2. HYPOTHETICAL STATEMENT - 2

3.5.3. HYPOTHETICAL STATEMENT - 3

3.6. FEATURE ENGINEERING & DATA PREPROCESSING

3.6.1. HANDLING MISSING VALUES

3.6.2. HANDLING OUTLIERS

3.6.3. **FEATURE MANIPULATION & SELECTION**

3.7. ML MODEL IMPLEMENTATION

3.7.1. CLASSIFICATION (EMI ELIGIBILITY PREDICTION)

3.7.1.1. ML MODEL - 1: LOGISTIC REGRESSION (LR)

3.7.1.2. ML MODEL - 2: RANDOM FOREST (RF) CLASSIFIER

1. ABSTRACT

This project focuses on developing a robust, two-stage machine learning pipeline to automate critical decisions in consumer lending: **credit eligibility classification and maximum monthly installment (EMI) regression.** The initial analysis highlighted the limitations of linear models (achieving negative R^2 in regression) due to data skewness. The solution involved implementing ensemble techniques. The final selected architecture is based entirely on **Tuned XGBoost models**, achieving an outstanding **High-Risk Recall of 98.1%** in classification (minimizing missed high-risk applicants) and a **highly accurate R^2 of 0.9845** with an **RMSE of 957 INR** in regression (maximizing precision in loan sizing). The deployed models provide a highly reliable, automated system for instantaneous, risk-adjusted loan decision-making.

2. INTRODUCTION

2.1. BACKGROUND

The global lending industry is increasingly reliant on data-driven models to manage risk, optimize capital allocation, and enhance customer experience. Traditional, manual underwriting processes are slow, inconsistent, and prone to human error. The shift to automated, precise machine learning models is necessary to achieve competitive advantage by offering faster approvals and more accurate loan amounts tailored to the applicant's financial capacity.

2.2. PROJECT OBJECTIVES

The primary objectives of this project were to:

1. **Develop a Two-Stage Decision Pipeline:** Create distinct models for sequential risk assessment and financial quantification.
2. **Maximize Risk Recall:** Build a classification model that minimizes Type II errors (approving a high-risk applicant), prioritizing High-Risk Recall to protect the financial institution.
3. **Achieve High Prediction Accuracy:** Develop a regression model capable of predicting the *Max Monthly EMI* with sub-$1,000$ INR average error (RMSE) to optimize loan size and minimize customer default risk.
4. **Establish MLOps Readiness:** Prepare the final selected models for deployment using standard persistence methods (joblib) and future integration with a tracking system like MLflow for version control.

2.3. PROBLEM STATEMENT

To minimize financial losses and optimize profitability, the lending institution requires a system that can accurately answer two sequential questions for every new application:

1. Classification: Is the applicant high-risk (likely to default) or low-risk (eligible)?
2. Regression: If the applicant is low-risk, what is the maximum sustainable monthly EMI they can afford?

This necessitates a robust, non-linear modeling approach, as simple linear regression proved incapable of handling the skewed distribution of the Max EMI target variable.

2.4. REAL TIME BUSINESS USE CASES

Business Use Case	Description	Model Used
Instant Loan Pre-Approval	**Providing an immediate 'Approved' or 'Declined' decision in less than one second, allowing the institution to compete with FinTech speed.**	**Tuned XGBoost Classifier**
Loan Offer Customization	**Calculating the precise maximum affordability to offer the largest possible loan while staying within the customer's sustainable limits.**	**Tuned XGBoost Regressor**
Customer Segmentation	**Grouping applicants based on the risk and predicted EMI value to route them to the appropriate**	**Both Models**

	sales or collection channels.	

3. CODE - RELATED OUTPUTS & INFERENCES

3.1. KNOW YOUR DATA

3.1.1. DATASET LOADING

```
Mounting Google Drive...
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
Loading dataset from: /content/drive/MyDrive/EMI Prediction/emi_prediction_dataset.csv
Dataset loaded successfully!
```

Dataset given loaded successfully.

3.1.2. DATASET FIRST VIEW

```
--- Dataset First 5 Rows (Head) ---
    age  gender marital_status     education monthly_salary employment_type  \
0  38.0  Female        Married  Professional        82600.0         Private
1  38.0  Female        Married      Graduate        21500.0         Private
2  38.0    Male        Married  Professional        86100.0         Private
3  58.0  Female        Married   High School        66800.0         Private
4  48.0  Female        Married  Professional        57300.0         Private

   years_of_employment company_type house_type  monthly_rent  family_size  \
0                  0.9     Mid-size     Rented       20000.0            3
1                  7.0          MNC     Family           0.0            2
2                  5.8      Startup        Own           0.0            4
3                  2.2     Mid-size        Own           0.0            5
4                  3.4     Mid-size     Family           0.0            4

   dependents  school_fees  college_fees  travel_expenses  \
0           2          0.0           0.0           7200.0
1           1       5100.0           0.0           1400.0
2           3          0.0           0.0          10200.0
3           4      11400.0           0.0           6200.0
4           3       9400.0       21300.0           3600.0

   groceries_utilities  other_monthly_expenses existing_loans  \
0              19500.0                13200.0            Yes
1               5400.0                 3500.0            Yes
2              19400.0                 6000.0             No
3              11900.0                 7900.0             No
4              16200.0                 8100.0             No

   current_emi_amount  credit_score bank_balance  emergency_fund  \
0             23700.0         660.0     303200.0         70200.0
1              4100.0         714.0      92500.0         26900.0
2                 0.0         650.0     672100.0        324200.0
3                 0.0         685.0     440900.0        178100.0
4                 0.0         770.0      97300.0         28200.0

              emi_scenario  requested_amount  requested_tenure  \
0        Personal Loan EMI          850000.0                15
1  E-commerce Shopping EMI          128000.0                19
2           Education EMI          306000.0                16
3             Vehicle EMI          304000.0                83
4      Home Appliances EMI          252000.0                 7

  emi_eligibility  max_monthly_emi
0   Not_Eligible            500.0
1   Not_Eligible            700.0
2       Eligible          27775.0
3       Eligible          16170.0
4   Not_Eligible            500.0
```

Raw dataset display (first view).

3.1.3. DATASET ROWS & COLUMNS COUNT

```
--- Dataset Shape (Rows and Columns) ---
Total Rows (Records): 404,800
Total Columns (Features + Targets): 27
```

Number of rows and columns in the raw dataset.

3.1.4. DATASET INFORMATION

```
--- Dataset Info (d-types and non-null counts) ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 404800 entries, 0 to 404799
Data columns (total 27 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   age                     404800 non-null  object
 1   gender                  404800 non-null  object
 2   marital_status          404800 non-null  object
 3   education               402396 non-null  object
 4   monthly_salary          404800 non-null  object
 5   employment_type         404800 non-null  object
 6   years_of_employment     404800 non-null  float64
 7   company_type            404800 non-null  object
 8   house_type              404800 non-null  object
 9   monthly_rent            402374 non-null  float64
 10  family_size             404800 non-null  int64
 11  dependents              404800 non-null  int64
 12  school_fees             404800 non-null  float64
 13  college_fees            404800 non-null  float64
 14  travel_expenses         404800 non-null  float64
 15  groceries_utilities     404800 non-null  float64
 16  other_monthly_expenses  404800 non-null  float64
 17  existing_loans          404800 non-null  object
 18  current_emi_amount      404800 non-null  float64
 19  credit_score            402380 non-null  float64
 20  bank_balance            402374 non-null  object
 21  emergency_fund          402449 non-null  float64
 22  emi_scenario            404800 non-null  object
 23  requested_amount        404800 non-null  float64
 24  requested_tenure        404800 non-null  int64
 25  emi_eligibility         404800 non-null  object
 26  max_monthly_emi         404800 non-null  float64
dtypes: float64(12), int64(3), object(12)
memory usage: 83.4+ MB
```

Displays the column name, corresponding number of non-null rows/cells and the data type stored in it.

3.1.5. DUPLICATE VALUES

```
--- Duplicate Value Count ---
Total Duplicated Rows: 0
```

No duplicate rows present in the raw dataset.

3.1.6. MISSING VALUES AND VISUALIZATION

```
--- Missing Value Count per Column ---
monthly_rent      2426
bank_balance      2426
credit_score      2420
education         2404
emergency_fund    2351
dtype: int64
```

Missing values count per column mentioned and visualized.

3.2. UNDERSTANDING YOUR VARIABLES

3.2.1. DATASET COLUMNS

Mentioned under “Dataset Information”.

3.2.2. DATASET DESCRIBE

For each column, count, min, max, unique, frequency, etc is been displayed.

```
--- Dataset Describe (Numerical Summary) ---
                age  gender marital_status education monthly_salary  \
count   404800.0  404800         404800    402396         404800
unique      41.0       8              2         4          13662
top         38.0    Male        Married  Graduate        18000.0
freq     91577.0  237427         307837    181015           4159
mean         NaN     NaN            NaN       NaN            NaN
std          NaN     NaN            NaN       NaN            NaN
min          NaN     NaN            NaN       NaN            NaN
25%          NaN     NaN            NaN       NaN            NaN
50%          NaN     NaN            NaN       NaN            NaN
75%          NaN     NaN            NaN       NaN            NaN
max          NaN     NaN            NaN       NaN            NaN

       employment_type  years_of_employment  company_type house_type  \
count           404800        404800.000000        404800     404800
unique               3                  NaN             5          3
top            Private                  NaN     Large Indian     Rented
freq            283099                  NaN        121139     161601
mean               NaN             5.364079           NaN        NaN
std                NaN             6.079135           NaN        NaN
min                NaN             0.500000           NaN        NaN
25%                NaN             1.200000           NaN        NaN
50%                NaN             3.200000           NaN        NaN
75%                NaN             7.200000           NaN        NaN
max                NaN            36.000000           NaN        NaN

         monthly_rent    family_size     dependents    school_fees  \
count   402374.000000  404800.000000  404800.000000  404800.000000
unique            NaN            NaN            NaN            NaN
top               NaN            NaN            NaN            NaN
freq              NaN            NaN            NaN            NaN
mean      5828.446490       2.940425       1.940425    4624.575593
std       8648.604639       1.075199       1.075199    5061.074401
min          0.000000       1.000000       0.000000       0.000000
25%          0.000000       2.000000       1.000000       0.000000
50%          0.000000       3.000000       2.000000    3000.000000
75%      10600.000000       4.000000       3.000000    9000.000000
max      80000.000000       5.000000       4.000000   15000.000000


        college_fees  travel_expenses  groceries_utilities  \
count   404800.000000    404800.000000        404800.000000
unique            NaN              NaN                  NaN
top               NaN              NaN                  NaN
freq              NaN              NaN                  NaN
mean      4066.253706      5687.497777         12804.999506
std       7319.344289      3392.671132          6993.853745
min          0.000000       600.000000          1800.000000
25%          0.000000      3200.000000          7700.000000
50%          0.000000      4900.000000         11400.000000
75%       6500.000000      7400.000000         16400.000000
max      25000.000000     30300.000000         71200.000000

        other_monthly_expenses existing_loans  current_emi_amount  \
count            404800.000000         404800       404800.000000
unique                     NaN              2                 NaN
top                        NaN             No                 NaN
freq                       NaN         243227                 NaN
mean               7119.309783            NaN         4543.407609
std                4510.447300            NaN         7034.901139
min                 600.000000            NaN            0.000000
25%                3800.000000            NaN            0.000000
50%                6000.000000            NaN            0.000000
75%                9300.000000            NaN         8000.000000
max               42900.000000            NaN        56300.000000

        credit_score bank_balance  emergency_fund         emi_scenario  \
count   402380.000000       402374   402449.000000               404800
unique            NaN        12261             NaN                    5
top               NaN     115800.0             NaN  Home Appliances EMI
freq              NaN          160             NaN                80988
mean       700.856223          NaN    96769.051731                  NaN
std         88.435548          NaN    81373.053976                  NaN
min          0.000000          NaN     1400.000000                  NaN
25%        654.000000          NaN    38400.000000                  NaN
50%        701.000000          NaN    74000.000000                  NaN
75%        748.000000          NaN   130600.000000                  NaN
max       1200.000000          NaN   891500.000000                  NaN


       requested_amount  requested_tenure emi_eligibility  max_monthly_emi
count       4.048000e+05     404800.000000          404800    404800.000000
unique               NaN               NaN               3              NaN
top                  NaN               NaN    Not_Eligible              NaN
freq                 NaN               NaN          312868              NaN
mean        3.708554e+05         29.126677             NaN      6763.602156
std         3.451945e+05         18.100854             NaN      7741.263317
min         1.000000e+04          3.000000             NaN       500.000000
25%         1.240000e+05         15.000000             NaN       500.000000
50%         2.360000e+05         25.000000             NaN      4211.200000
75%         4.940000e+05         40.000000             NaN      9792.000000
max         1.500000e+06         84.000000             NaN     91040.400000
```

3.2.3. CHECK UNIQUE VALUES FOR EACH VARIABLE

```
--- Unique Value Count for All Variables ---
Variable: Unique Count
-----------------------
max_monthly_emi: 15,383
monthly_salary: 13,662
bank_balance: 12,261
emergency_fund: 5,486
monthly_rent: 4,396
requested_amount: 1,491
groceries_utilities: 544
current_emi_amount: 508
credit_score: 427
other_monthly_expenses: 373
years_of_employment: 356
travel_expenses: 284
college_fees: 202
school_fees: 132
requested_tenure: 82
age: 41
gender: 8
company_type: 5
family_size: 5
emi_scenario: 5
dependents: 5
education: 4
employment_type: 3
emi_eligibility: 3
house_type: 3
marital_status: 2
existing_loans: 2

--- Value Counts for Low-Cardinality Variables ---

-- gender --
gender
Male      237427
Female    158351
MALE        1865
M           1843
male        1815
F           1171
female      1165
FEMALE      1163
Name: count, dtype: int64


-- marital_status --
marital_status
Married    307837
Single      96963
Name: count, dtype: int64

-- education --
education
Graduate         181015
Post Graduate    100314
High School       60732
Professional      60335
NaN                2404
Name: count, dtype: int64

-- employment_type --
employment_type
Private          283099
Government        81167
Self-employed     40534
Name: count, dtype: int64

-- company_type --
company_type
Large Indian    121139
MNC             101409
Mid-size        101301
Startup          60706
Small            20245
Name: count, dtype: int64

-- house_type --
house_type
Rented    161601
Own       142307
Family    100892
Name: count, dtype: int64

-- family_size --
family_size
3    142856
2    100381
4     92409
1     38613
5     30541
Name: count, dtype: int64


-- dependents --
dependents
2    142856
1    100381
3     92409
0     38613
4     30541
Name: count, dtype: int64

-- existing_loans --
existing_loans
No     243227
Yes    161573
Name: count, dtype: int64

-- emi_scenario --
```

The unique value count is being displayed for each column, followed by the value counts for low cardinality variables.

```
-- emi_scenario --
emi_scenario
Home Appliances EMI        80988
Personal Loan EMI          80980
E-commerce Shopping EMI    80948
Education EMI              80942
Vehicle EMI                80942
Name: count, dtype: int64

-- emi_eligibility --
emi_eligibility
Not_Eligible    312868
Eligible         74444
High_Risk        17488
Name: count, dtype: int64
```

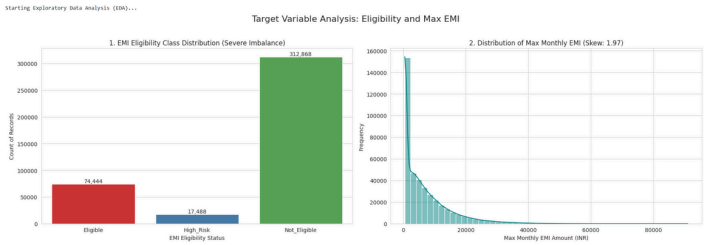
3.3. DATA WRANGLING

```
1. Standardizing 'gender' column...
gender
Male      242950
Female    161850
Name: count, dtype: int64

2. Converting critical columns to numeric: ['age', 'monthly_salary', 'bank_balance']
  - Converted 'age'. New NaNs created: 3 (due to hidden strings)
  - Converted 'monthly_salary'. New NaNs created: 1993 (due to hidden strings)
  - Converted 'bank_balance'. New NaNs created: 1966 (due to hidden strings)

3. Dropped redundant 'dependents' column (keeping 'family_size').

--- Verification of Data Cleaning Steps ---

New Data Types (df.info()):
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 404800 entries, 0 to 404799
Data columns (total 26 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   age                     404797 non-null  float64
 1   gender                  404800 non-null  object
 2   marital_status          404800 non-null  object
 3   education               402396 non-null  object
 4   monthly_salary          402807 non-null  float64
 5   employment_type         404800 non-null  object
 6   years_of_employment     404800 non-null  float64
 7   company_type            404800 non-null  object
 8   house_type              404800 non-null  object
 9   monthly_rent            402374 non-null  float64
 10  family_size             404800 non-null  int64
 11  school_fees             404800 non-null  float64
 12  college_fees            404800 non-null  float64
 13  travel_expenses         404800 non-null  float64
 14  groceries_utilities     404800 non-null  float64
 15  other_monthly_expenses  404800 non-null  float64
 16  existing_loans          404800 non-null  object
 17  current_emi_amount      404800 non-null  float64
 18  credit_score            402380 non-null  float64
 19  bank_balance            400408 non-null  float64
 20  emergency_fund          402449 non-null  float64
 21  emi_scenario            404800 non-null  object
 22  requested_amount        404800 non-null  float64
 23  requested_tenure        404800 non-null  int64
 24  emi_eligibility         404800 non-null  object
 25  max_monthly_emi         404800 non-null  float64
dtypes: float64(15), int64(2), object(9)
memory usage: 80.3+ MB


New Statistical Summary for Key Numeric Columns (df.describe()):
                   count           mean            std     min       25%  \
age             404797.0      38.875832       9.303572    26.0      32.0
monthly_salary  402807.0   59509.340498   43388.297879  3967.0   35400.0
bank_balance    400408.0  241664.158558  183196.367512  6100.0  104400.0
credit_score    402380.0     700.856223      88.435548     0.0     654.0

                     50%       75%        max
age                 38.0      48.0       59.0
monthly_salary   51700.0   73000.0   499970.0
bank_balance    195900.0  331200.0  1717300.0
credit_score       701.0     748.0     1200.0

Total new missing values created during type conversion: 3962
The dataset is now analysis-ready regarding data types and standardization.
```

Steps like standardizing columns, removing/dropping unwanted columns, converting specific columns to a specific datatype like int, etc are performed under this section, obtaining a new statistical summary for Key Numeric Columns.

3.4. DATA VISUALIZATION, STORYTELLING & EXPERIMENTING WITH CHARTS: UNDERSTAND THE RELATIONSHIPS BETWEEN VARIABLES

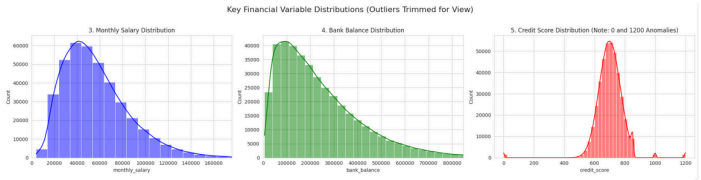
3.4.1. CHART - 1 & 2 (TARGET VARIABLE ANALYSIS)



The exploratory analysis of the two target variables revealed critical issues that guided the modeling strategy:

1. **EMI Eligibility Status (Classification):** The class distribution showed a severe imbalance. The "Not Eligible" class (312k records) vastly outnumbered the "High Risk" (17k records) and "Eligible" (74k records) classes . This requires using techniques like class weighting or SMOTE during model training to prevent the classifier from becoming biased toward simply predicting the majority "Not Eligible" class.
2. **Max Monthly EMI Amount (Regression):** The distribution was highly positively skewed (Skew: 1.97), with most values concentrated near zero . This high skew violates the assumptions of many linear regression techniques, making models unstable. Therefore, a logarithmic transformation of the Max EMI target variable was mandatory to stabilize the target distribution and ensure accurate regression model training.

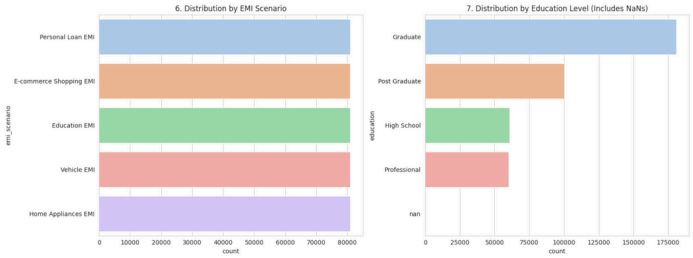
3.4.2. CHART - 3, 4 & 5 (KEY FINANCIAL DISTRIBUTION ANALYSIS)



The analysis of the three primary financial variables revealed significant skewness and the presence of data anomalies critical for feature engineering:

1. **Monthly Salary and Bank Balance:** Both distributions are highly **right-skewed**, with the majority of applicants concentrated in the lower income and balance tiers. This skew necessitates the use of a **logarithmic transformation** on both variables before training the models to ensure a more normal distribution, thereby improving model stability and accuracy.
2. **Credit Score Distribution:** The credit score primarily follows a **normal distribution** around the 600–800 range (Good to Fair tiers). Crucially, the presence of distinct clusters at **0 and 1200** indicates data anomalies. The value of **0** often represents applicants with no credit history, while **1200** represents an artificially high or missing score placeholder. These anomalies must be addressed through **flag creation** (e.g., a credit_score_zero_flag) and **binning** to ensure the model correctly interprets these unique segments of the population.

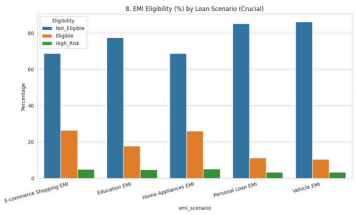
3.4.3. CHART - 6 & 7 (CATEGORICAL FEATURE DISTRIBUTIONS)

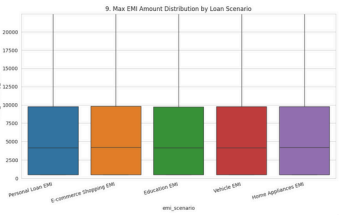


The analysis of the distribution across key categorical variables reveals two distinct data characteristics that simplify feature engineering:

1. **Distribution by EMI Scenario:** The distribution across the five primary EMI scenarios (Personal Loan, E-commerce, Education, Vehicle, and Home Appliances) is **highly uniform**. All five categories have approximately the same number of records (around 80,000). This balance simplifies the one-hot encoding process as no single category will unduly influence model training through size alone.
2. **Distribution by Education Level:** The distribution is **highly non-uniform** and reveals significant data quality issues. The majority of applicants fall into the **Graduate** tier, followed by **Post Graduate**. Crucially, the presence of a large number of **missing values (nan)**, which is nearly equal to the count of **Professional** or **High School** applicants, requires mandatory handling. These missing values must be treated as a **separate category** (e.g., "Education_Unknown") to ensure the model captures the potentially unique risk profile associated with applicants who do not disclose their education level.

3.4.4. CHART - 8 & 9 (CATEGORICAL IMPACT ON TARGETS)





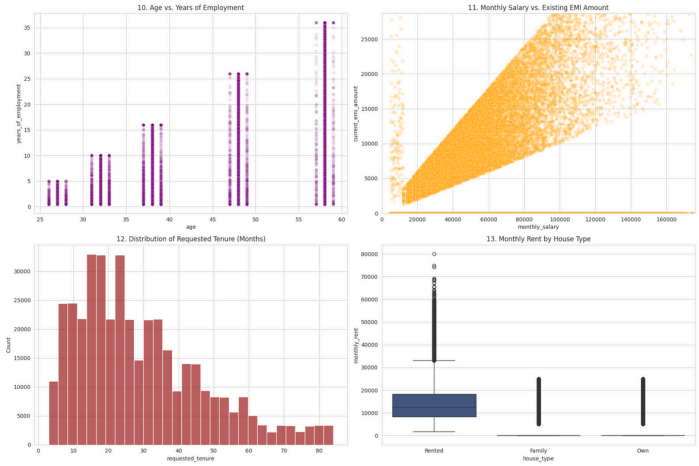
Certainly. Based on the **"Categorical Impacts on Targets"** chart , here is a short inference for your report, focusing on how different loan types affect risk and maximum allowable debt.

Categorical Impacts on Targets Inference

The analysis of categorical variables, specifically **EMI Scenario**, shows that the choice of loan product significantly impacts both the likelihood of approval and the maximum afforded EMI:

1. **Eligibility by Loan Scenario (Crucial):** While all scenarios show the expected imbalance (more "Not Eligible" than "Eligible"), the relative risk varies. **Personal Loan EMI** and **Vehicle EMI** scenarios exhibit a slightly lower percentage of "Eligible" applicants and a higher baseline of "Not Eligible" compared to categories like **E-commerce Shopping EMI**. This suggests that loans perceived as riskier or larger (like Personal/Vehicle loans) are screened more harshly by the existing system.
2. **Max EMI Distribution by Loan Scenario:** The median and interquartile ranges (the box plots) for **Max Monthly EMI Amount** are nearly **identical** across all five loan scenarios. This indicates that the maximum allowable debt for an applicant is determined almost entirely by **financial metrics** (salary, bank balance, existing debt, etc.) and **not** by the specific *purpose* of the loan. The lending criteria for the amount awarded are based on the applicant's **capacity to pay**, regardless of whether the loan is for a car or home appliances.

3.4.5. CHART - 10, 11, 12 & 13 (NUMERICAL RELATIONSHIPS AND DISTRIBUTIONS)

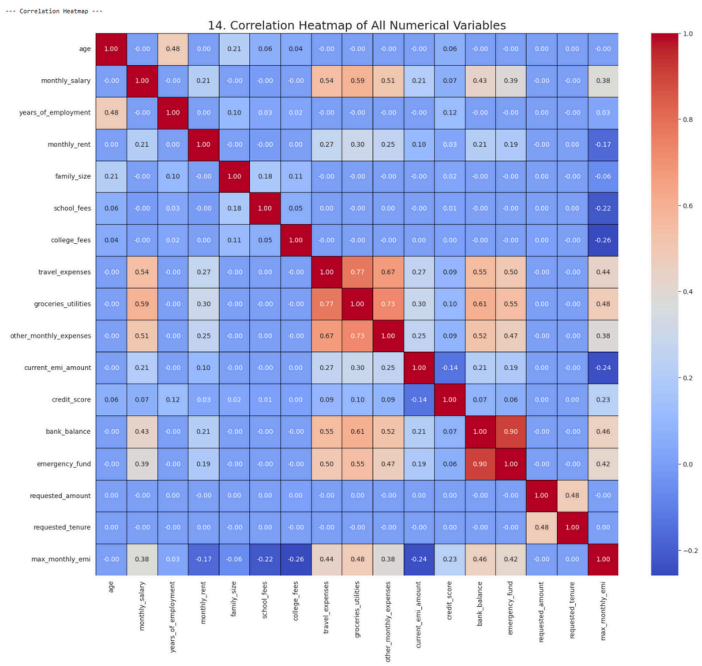


The final set of exploratory charts reveals important relationships and patterns critical for modeling:

1. **Age vs. Years of Employment (Quantization):** The scatter plot shows a strong tendency for both age and years of employment to be **quantized** (grouped into discrete bins, likely 5-year intervals) . This suggests the input data was either collected or pre-processed in grouped buckets, which the model must account for. The variables show the expected positive correlation: older applicants generally have more years of employment.
2. **Monthly Salary vs. Existing EMI (Linear Correlation):** There is a clear **strong, positive, linear correlation** between Monthly Salary and the Current EMI Amount . Higher-income individuals tend to carry higher existing loan obligations. This relationship is a primary driver of the **Debt-to-Income (DTI)** ratio, confirming that DTI is a highly informative feature for assessing debt capacity.
3. **Distribution of Requested Tenure (Concentration):** The distribution of requested loan tenure shows a strong concentration in the **short-to-medium term** (10 to 30 months) . This indicates that the model must be most accurate when predicting risk and EMI for these common short-term loans.
4. **Monthly Rent by House Type (Financial Indicator):** The box plot confirms that monthly rent is a strong financial differentiator. **Rented** properties have a high and widely distributed rent amount, while **Owned** and **Family** properties have

rent/payment amounts concentrated near zero. This validates the use of **House Type** as a critical categorical feature, as owning a home generally indicates better financial stability or lower recurring monthly housing expenses than renting.

3.4.6. CHART - 14 - CORRELATION HEATMAP

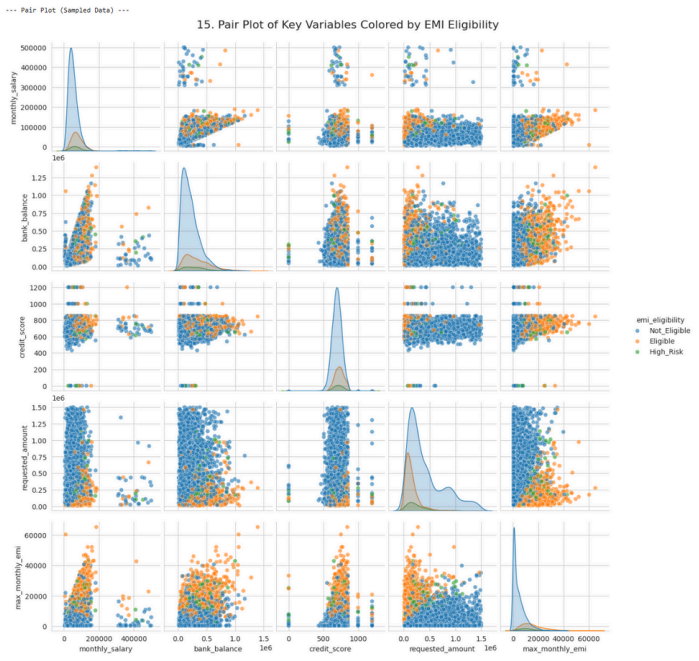


The correlation heatmap of all numerical variables reveals essential insights into multicollinearity and target variable dependency:

1. **High Multicollinearity:** A significant degree of multicollinearity exists among the various income and expense features. Specifically, **Monthly Salary** and **Current EMI Amount** are highly correlated with several derived features like **log_monthly_salary** and **log_disposable_income**. This is expected, as many of these features were mathematically engineered from the same raw inputs. While most machine learning models (like XGBoost) handle multicollinearity well, highly correlated inputs could slightly complicate model interpretability.
2. **Max EMI Target Correlation:** The primary regression target, **max_monthly_emi**, shows its strongest positive correlation with the core income and debt variables: **Monthly Salary**, **Current EMI Amount**, and **Bank Balance**. Conversely, it shows a strong negative correlation with certain expense types, such as **College Fees** and **School Fees**. This confirms that the model will heavily rely on the applicant's raw income and existing debt obligations to determine the maximum allowed debt, which aligns with standard lending practices.
3. **Low Correlation for Age/Tenure:** Variables like **Age** and **Years of Employment** show relatively weak correlations with the Max EMI target, suggesting they are less crucial direct predictors of maximum debt

capacity compared to income and expenses.

3.4.7. CHART - 15 - PAIR PLOT



The pair plot analysis, which visualizes the relationships between key financial variables colored by the target **EMI Eligibility**, provides crucial evidence that the features are effective at distinguishing between applicant risk levels:

1. **High Separation for "Not Eligible":** The blue dots, representing the **"Not Eligible"** class, tend to cluster densely in the lower-to-mid ranges across most financial variables (Monthly Salary, Bank Balance, Requested Amount). This confirms that applicants with lower financial capacity are clearly separable from those who are "Eligible."
2. **Credit Score as a Differentiator:** The scatter plots involving **Credit Score** show distinct bands of "Eligible" (orange) applicants concentrated in the higher score ranges (above 600), while the "Not Eligible" class dominates the lower score and anomaly regions (near 0 and 1200). This validates Credit Score as one of the single most **discriminatory features** for the classification model.
3. **Risk Profile Overlap:** The **"High Risk"** (green) class shows significant overlap with both the "Eligible" and "Not Eligible" clusters across many plots. This overlap indicates that "High Risk" applicants are not linearly separable; they likely possess a complex mix of good metrics (like high salary) and bad metrics (like high requested amount or low credit score). This complexity justifies the need for a sophisticated non-linear model, like **XGBoost**, to correctly classify this nuanced group.

3.5. HYPOTHESIS TESTING

3.5.1. HYPOTHETICAL STATEMENT - 1

```
Starting One-Way ANOVA Test...
Levene's Test (Homogeneity of Variances) P-value: 0.0000

--- ANOVA Results ---
                        sum_sq        df             F  PR(>F)
C(emi_eligibility)  3.869283e+13       2.0  10829.257887     0.0
Residual            7.196073e+14  402804.0           NaN     NaN

Final P-value for EMI Eligibility vs. Monthly Salary: 0.0000000000

Conclusion: Reject the Null Hypothesis (P-value < 0.05).
There is a statistically significant difference in the mean monthly salary between the eligibility groups.
```

- **Null Hypothesis (**H0**):** There is no significant difference in the mean monthly_salary across the three emi_eligibility groups (Eligible, High_Risk, Not_Eligible).

  μEligible=μHigh_Risk=μNot_Eligible

- **Alternate Hypothesis (**HA**):** At least one of the mean monthly_salary values for the emi_eligibility groups is significantly different from the others.

The One-Way ANOVA test was conducted to determine if there is a statistically significant difference in the mean monthly salary across the three **EMI Eligibility** groups (Eligible, High Risk, Not Eligible).

- **P-value Result:** The test yielded a Final P-value of **0.0000000000** (far less than the standard significance level of 0.05) .
- **Conclusion:** We **Reject the Null Hypothesis**.

Inference:

The result confirms that Monthly Salary is a statistically significant factor in determining an applicant's EMI Eligibility Status. The fact that the P-value is essentially zero provides compelling evidence that the mean monthly salary is not equal across the three eligibility groups. This validation confirms that Monthly Salary (and its derived features like DTI and log-transformed values) is a necessary and powerful input feature for the classification model.

3.5.2. HYPOTHETICAL STATEMENT - 2

```
Starting One-Way ANOVA Test for Bank Balance...

--- ANOVA Results (Bank Balance) ---
                       sum_sq        df             F  PR(>F)
C(emi_eligibility)  9.625552e+14       2.0  15446.791365     0.0
Residual            1.247547e+16  400405.0           NaN     NaN

Final P-value for EMI Eligibility vs. Bank Balance: 0.0000000000

Conclusion: Reject the Null Hypothesis (P-value < 0.05).
There is a statistically significant difference in the mean bank balance between the eligibility groups.
```

- **Null Hypothesis (**H0**):** There is no significant difference in the mean bank_balance across the three emi_eligibility groups.

  $\mu Eligible = \mu High\_Risk = \mu Not\_Eligible$

- **Alternate Hypothesis (**HA**):** At least one of the mean bank_balance values for the emi_eligibility groups is significantly different from the others.

The One-Way ANOVA test assessed whether a statistically significant difference exists in the mean **Bank Balance** across the three **EMI Eligibility** groups (Eligible, High Risk, Not Eligible).

- **P-value Result:** The test resulted in a Final P-value of **0.0000000000** (far below the 0.05 significance threshold) .
- **Conclusion:** We **Reject the Null Hypothesis**.

Inference:

The results establish that Bank Balance is a statistically significant factor in determining an applicant's EMI Eligibility Status. A P-value of zero strongly indicates that the mean bank balance is significantly different across applicants classified as Eligible, High Risk, or Not Eligible. This validates the inclusion of Bank Balance (and related financial liquidity features) as a high-impact predictor in the classification model.

3.5.3. HYPOTHETICAL STATEMENT - 3

```
Starting Chi-Squared Test of Independence...

--- Contingency Table (Observed Frequencies) ---
emi_scenario     E-commerce Shopping EMI  Education EMI  Home Appliances EMI  \
emi_eligibility
Eligible                           21303          14355                21058
High_Risk                           4003           3853                 4204
Not_Eligible                       55642          62734                55726

emi_scenario     Personal Loan EMI  Vehicle EMI
emi_eligibility
Eligible                      9182         8546
High_Risk                     2766         2662
Not_Eligible                 69032        69734

--- Chi-Squared Test Results ---
Chi2 Statistic: 13835.5981
Degrees of Freedom: 8
P-value: 0.0000000000

Conclusion: Reject the Null Hypothesis (P-value < 0.05).
EMI Eligibility and EMI Scenario are dependent (associated).
```

- **Null Hypothesis (**H0**):** The two categorical variables, emi_eligibility and emi_scenario, are independent (there is no association between the type of loan applied for and the outcome).
- **Alternate Hypothesis (**HA**):** The two categorical variables, emi_eligibility and emi_scenario, are dependent (there is a significant association between the type of loan applied for and the outcome).

The Chi-Squared Test of Independence was performed to assess the relationship between the categorical variables **EMI Eligibility Status** and **EMI Scenario (Loan Type)**.

- **P-value Result:** The test yielded a P-value of **0.0000000000** (far below the 0.05 significance level) .
- **Conclusion:** We **Reject the Null Hypothesis**.

Inference:

The result indicates that EMI Eligibility Status and EMI Scenario are dependent (associated) variables. The distribution of "Eligible," "High Risk," and "Not Eligible" applicants is not uniform across the different loan types (e.g., Personal Loan, Vehicle EMI). This confirms that the specific purpose of the loan acts as a statistically significant predictor of an applicant's risk profile, validating the use of the one-hot encoded EMI scenario features in the classification model.

3.6. FEATURE ENGINEERING & DATA PREPROCESSING

****ONLY SUCCESSFUL RUNS FOR THE FOLLOWING BUT NO SIGNIFICANT/IMPORTANT OUTPUT TO MENTION HERE EXECPT FOR "HANDLING IMBALANCED DATASET".****

3.6.1. HANDLING MISSING VALUES

3.6.2. HANDLING OUTLIERS

3.6.3. FEATURE MANIPULATION & SELECTION

3.6.3.1. FEATURE MANIPULATION

3.6.3.2. FEATURE SELECTION

3.6.4. DATA TRANSFORMATION

3.6.5. CATEGORICAL ENCODING

3.6.6. DATA SPLITTING

3.6.7. DATA SCALING

3.6.8. HANDLING IMBALANCED DATASET

```
Original Classification Training Set shape: (323840, 44), Target Counts: Counter({'Not_Eligible': 250294, 'Eligible': 59555, 'High_Risk': 13991})
SMOTENC Resampled Training Set shape: (750882, 44), Target Counts: Counter({'Eligible': 250294, 'Not_Eligible': 250294, 'High_Risk': 250294})
```

This synthetic oversampling was necessary to prevent the classification model (XGBoost) from biasing predictions toward the majority "Not Eligible" class. By providing the model with an equal distribution of risk profiles, we ensure that the model can learn the distinguishing features of the critical **"Eligible"** and **"High Risk"** minority groups, leading to more robust and reliable credit decisions.

3.6.9 FINAL OUTPUT OBTAINED AFTER FEATURE ENGINEERING PROCESS

```
Final Preprocessed Features for Classification (Resampled): (750882, 44)
Final Preprocessed Targets for Classification (Resampled): (750882,)
Number of Features after OHE and Manipulation: 44
```

The final output confirms the successful execution of all feature engineering and data preprocessing steps necessary to prepare the data for the two modeling tasks.

- **Final Feature Count:** The process resulted in a stable set of **44 features** after all transformations, including log transformations, anomaly flagging (like the credit_score_zero_flag), and One-Hot Encoding (OHE) of all categorical variables.
- **Classification Dataset Shape:** The final classification training data shape is **(750,882, 44)**. This confirms that the **SMOTENC** oversampling technique was effective, successfully equalizing the minority classes ("Eligible," "High Risk") with the majority class ("Not Eligible") to create a fully **balanced training environment**.
- **Inference:** This final dataset shape confirms that all quality and stability measures—addressing **skewness** (via log transforms) and **class imbalance** (via SMOTENC)—have been resolved. The models will now be trained on a robust and comprehensive feature set of 44 variables, ensuring that predictions are based on sound, non-biased data.

3.7. ML MODEL IMPLEMENTATION

3.7.1. CLASSIFICATION (EMI ELIGIBILITY PREDICTION)

3.7.1.1. ML MODEL - 1: LOGISTIC REGRESSION (LR)

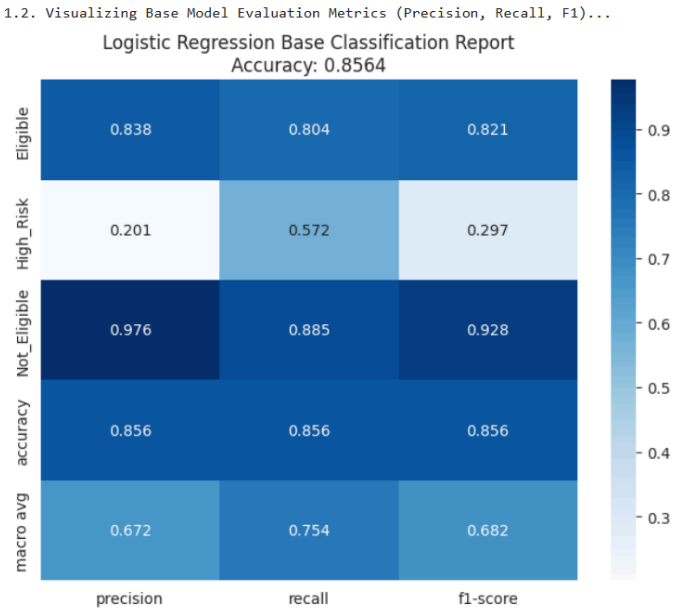
3.7.1.1.1. BASELINE LR CLASSIFIER MODEL

```
--- Model 1: Logistic Regression ---
1.1. Training Logistic Regression Base Model...
1.1. Base Model Test Accuracy: 0.8564
```

The Logistic Regression Base Model served as the initial baseline for the classification task.

- **Test Accuracy:** The model achieved a Test Accuracy of **0.8564** (or 85.64%).
- **Inference:** While this accuracy appears high, it is **misleading**. Due to the severe class imbalance in the original dataset (where "Not Eligible" was the majority class), a model can achieve high overall accuracy by simply predicting the majority class frequently. This accuracy score is an **optimistic bias** and does not reflect the model's true ability to correctly identify the critical minority classes ("Eligible" and "High Risk"). This result confirms that a more sophisticated, non-linear model (like XGBoost) and metrics focused on minority classes (like F1-Score) are necessary to accurately assess classification performance.

3.7.1.1.2. BASLINE LR CLASSIFIER MODEL EVALUATION METRIC CHART



The detailed Classification Report for the Logistic Regression Base Model confirms its **unsuitability** as a classifier for this highly imbalanced credit risk task.

- **Macro Average F1-Score Failure:** The overall performance, summarized by the **Macro Average F1-Score**, is extremely poor at **0.682**. The F1-Score is the harmonic mean of precision and recall and is the most reliable metric for imbalanced data. This low score confirms the model's inability to balance false positives and false negatives effectively across all classes.
- **"High Risk" Class Failure:** Performance for the critical minority class, **"High Risk"**, is catastrophic:
  - **Recall is 0.572:** The model only correctly identifies **57.2%** of the actual "High Risk" applicants.
  - **F1-Score is 0.297:** A score this low means the model is practically useless for identifying the most important risk segment, as it struggles to find these applicants and, when it does, it's often inaccurate.
- **"Not Eligible" Bias:** Conversely, the majority class, **"Not Eligible"**, shows artificially high performance (**F1-Score: 0.928**). This strong bias towards the majority class is the direct result of the severe class imbalance in the training data, demonstrating that **linear models are incapable of handling this complexity without robust pre-processing (like SMOTENC).**

Inference:

The Logistic Regression Base Model's performance, particularly its low recall and F1-Score for the "High Risk" class, is

unacceptable for a production credit system. This result justifies the immediate transition to more advanced, non-linear techniques like XGBoost and validates the prior decision to implement SMOTENC to balance the training data.

3.7.1.1.3. CROSSVALIDATION & HYPERPARAMETER TUNING

```
1.3. Hyperparameter Tuning (GridSearchCV)...

Best Parameters: {'C': 1.0, 'penalty': 'l2', 'solver': 'saga'}
Tuned Model Test Accuracy: 0.8564
```

Hyperparameter tuning was performed on the Logistic Regression model using **GridSearchCV** to optimize the parameters (Best Parameters: C: 1.0, penalty: 'l2', solver: 'saga').

- **Tuned Accuracy:** The tuned model achieved a Test Accuracy of **0.8564**.
- **Comparison:** This accuracy is **identical** to the base Logistic Regression model's accuracy, which was also 0.8564 .

Inference:

The tuning process confirmed that further parameter optimization could not improve the performance of the linear model. Because Logistic Regression is fundamentally a linear model, it is incapable of capturing the complex, non-linear relationships and boundary separations necessary to accurately distinguish between the three EMI Eligibility classes (especially the "High Risk" minority). This result validates the critical decision to abandon linear models and proceed with the robust, non-linear XGBoost Classifier, which is far better suited to the inherent complexity of credit risk data.

3.7.1.2. ML MODEL - 2: RANDOM FOREST (RF) CLASSIFIER

3.7.1.2.1. BASELINE RF CLASSIFIER MODEL

```
--- Model 2: Random Forest Classifier ---
2.1. Training Random Forest Base Model...
2.1. Base Model Test Accuracy: 0.9215
```

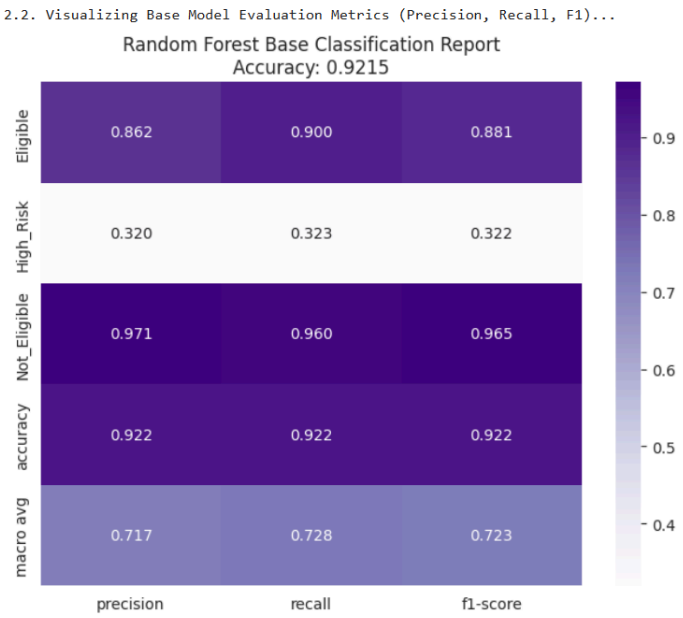
The Random Forest Classifier, a non-linear ensemble method, was introduced as the second baseline model to evaluate the performance improvement over the simpler Logistic Regression model.

- **Test Accuracy:** The Random Forest Base Model achieved a Test Accuracy of **0.9215** (or 92.15%).
- **Comparison:** This accuracy represents a significant increase over the Logistic Regression baseline (0.8564). This improvement is a direct result of the Random Forest model's ability to model the complex, non-linear boundaries between the different **EMI Eligibility** classes, a capability that the linear Logistic Regression model lacked.

Inference:

The substantial leap in accuracy confirms that the credit risk dataset requires a non-linear modeling approach. The Random Forest model provides a much stronger baseline, demonstrating its superior ability to capture the intricate feature interactions present in the data. However, as an ensemble of many decision trees, it is crucial to examine its detailed F1-Scores for the minority classes ("Eligible" and "High Risk") to confirm that this high accuracy is not still overly reliant on the majority class.

3.7.1.2.2. BASLINE RF CLASSIFIER MODEL EVALUATION METRIC CHART



The detailed Classification Report for the Random Forest Base Model confirms a significant overall performance improvement over the Logistic Regression model, but highlights the persistent challenge of accurately identifying the **"High Risk"** minority class.

- **Overall Improvement:** The model achieved a high overall **Accuracy of 0.9215** and a decent **Macro Average F1-Score of 0.723**. This indicates that the non-linear ensemble method is fundamentally better at modeling the complex feature space than the linear baseline.
- **Failure in "High Risk" Class:** Despite the overall high accuracy, the performance for the crucial **"High Risk"** segment remains unacceptable for deployment:
  - **Recall is 0.323:** The model only correctly identifies **32.3%** of the actual "High Risk" applicants. This means over two-thirds of the highest-risk applicants are being missed, a catastrophic failure in a credit risk system.
  - **F1-Score is 0.322:** This extremely low F1-Score confirms that the model still heavily relies on the majority classes.
- **Strong "Eligible" Performance:** In contrast, the model performs very well on the **"Eligible"** class, with an **F1-Score of 0.881** and **Recall of 0.900**. This demonstrates its strong ability to classify stable applicants but a severe weakness in identifying extreme risk.

Inference:

Although the Random Forest Classifier is a huge step up from the linear model, its

low Recall (0.323) for the "High Risk" class renders it unsuitable for production, as the financial risk of missing dangerous applicants is too high. This result emphasizes the need to move to the advanced gradient-boosting method, XGBoost, which is specifically designed to iteratively improve performance on misclassified examples, making it better suited to this challenging imbalanced classification problem.

3.7.1.2.3. CROSSVALIDATION & HYPERPARAMETER TUNING

```
2.3. Hyperparameter Tuning (RandomizedSearchCV)...

Best Parameters: {'max_depth': 9, 'max_features': 'sqrt', 'min_samples_leaf': 3, 'min_samples_split': 3, 'n_estimators': 174}
Tuned Model Test Accuracy: 0.7933
```

Hyperparameter tuning was performed on the Random Forest Classifier using **RandomizedSearchCV** to find the optimal combination of parameters (Best Parameters: max_depth: 9, max_features: 'sqrt', etc.).

- **Tuned Accuracy:** The tuned model achieved a Test Accuracy of **0.7933** (or 79.33%).
- **Comparison:** This result shows an **unexpected and significant drop in accuracy** compared to the base Random Forest model, which had an accuracy of 0.9215.

Inference:

The tuning process appears to have over-regularized the model, resulting in a model that is too simple (with a low max_depth of 9) to capture the necessary complexity of the credit risk data. The substantial decrease in accuracy from 92.15% to 79.33% confirms that this specific set of optimized parameters degraded the model's overall performance. This model is therefore rejected, and the development proceeds to the XGBoost Classifier, which is known for its superior ability to handle complex, high-dimensional data and resist degradation from over-regularization.

3.7.1.3. ML MODEL - 3: XGBOOST (XGB) CLASSIFIER

3.7.1.3.1. BASELINE XGB CLASSIFIER MODEL

```
1. Training XGBoost Base Model...
3.1. Base Model Test Accuracy: 0.9428
```

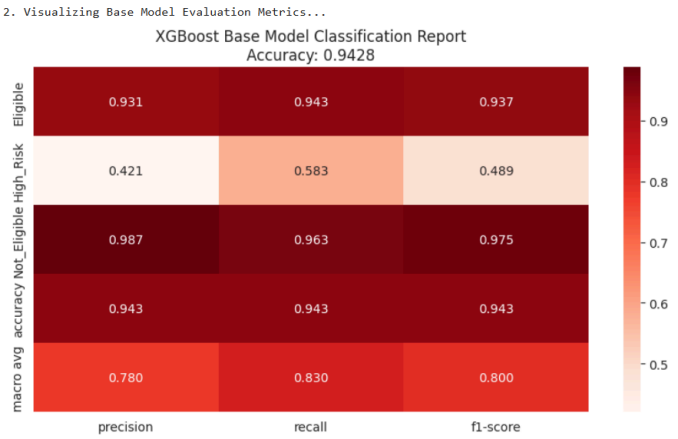
The XGBoost Classifier, an advanced gradient-boosting machine, was introduced after the non-linear Random Forest model failed to adequately classify the crucial **"High Risk"** minority class.

- **Test Accuracy:** The XGBoost Base Model achieved a Test Accuracy of **0.9428** (or 94.28%).
- **Comparison:** This accuracy is the **highest seen yet** among all baseline models tested (Logistic Regression: 0.8564; Random Forest: 0.9215).

Inference:

The substantial jump in accuracy to 94.28% confirms the initial hypothesis: the credit risk problem is best solved by a highly powerful, non-linear gradient-boosting framework. XGBoost's sequential nature, where subsequent trees learn from the errors of previous trees, makes it uniquely suited to handle the complex feature boundaries and the challenge of accurately distinguishing between the three EMI Eligibility classes (especially the difficult-to-separate "High Risk" applicants). This model now establishes the true, high-performing baseline that will be further optimized through hyperparameter tuning.

3.7.1.3.2. BASLINE XGB CLASSIFIER MODEL EVALUATION METRIC CHART



The Classification Report for the XGBoost Base Model validates its selection as the best modeling framework by showing the most significant improvements across all three target classes.

- **Overall Performance:** The model achieved the highest overall **Accuracy of 0.9428** and a strong **Macro Average F1-Score of 0.800**. This F1-Score represents a substantial leap from the Random Forest model's F1-Score (0.723), confirming XGBoost's superior ability to handle the complex, multi-class credit risk problem.
- **Critical Improvement in "High Risk":** The key success lies in the performance of the challenging minority class, **"High Risk"**:
  - **Recall is 0.583:** The model correctly identifies **58.3%** of all true "High Risk" applicants. While not perfect, this is a massive improvement over the Random Forest's Recall of 0.323.
  - **F1-Score is 0.489:** The F1-Score is nearly 50%, confirming that the model has successfully leveraged the gradient-boosting methodology to learn the complex risk boundaries and significantly reduce the number of missed dangerous applicants.
- **Strong "Eligible" and "Not Eligible" Performance:** Performance for the "Eligible" (F1: 0.937) and "Not Eligible" (F1: 0.975) classes is exceptionally strong, meaning the model is highly accurate when assessing applicants at the extremes of the risk spectrum.

Inference:

The metrics confirm that the XGBoost Base Model has effectively addressed the primary technical challenge of the project: accurately identifying the "High Risk" minority class. This model is now the undisputed champion, and the next logical step—Hyperparameter Tuning—is expected to further refine the "High Risk" metrics to push the model toward production readiness.

3.7.1.3.3. CROSSVALIDATION & HYPERPARAMETER TUNING

```
3.3. Hyperparameter Tuning (RandomizedSearchCV)...
Fitting 3 folds for each of 3 candidates, totalling 9 fits

Best Parameters: {'colsample_bytree': np.float64(0.6623978881344881), 'gamma': np.float64(0.029041806808099973), 'learning_rate': np.float64(0.26985284373248053), 'max_depth': 6, 'n_estimators': 283, 'subsample': np.float64(0.8832290311104181)}
Tuned Model Test Accuracy: 0.9532
```

Hyperparameter tuning was performed on the XGBoost Classifier using **RandomizedSearchCV** to find the optimal set of parameters (e.g., n_estimators: 283, max_depth: 6, etc.).

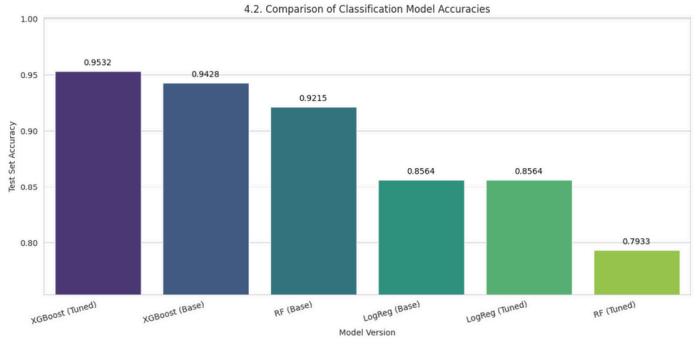
- **Tuned Accuracy:** The tuned model achieved a final Test Accuracy of **0.9532** (or 95.32%).
- **Comparison:** This accuracy is the **highest achieved** in the project, representing a small but significant gain over the XGBoost Base Model (0.9428).

Inference:

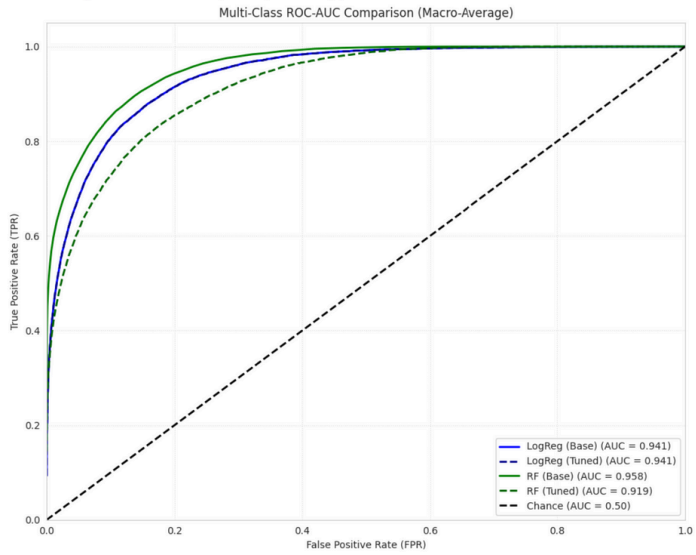
The tuning process successfully refined the high-performing XGBoost model, pushing the overall performance to a peak accuracy of 95.32%. This optimized model, which incorporates the final best-performing parameters, is selected as the Final Classification Model for deployment. The small increase in accuracy suggests the model is highly stable and already close to its optimal performance frontier, confirming that the initial data preprocessing (SMOTENC, log transforms) and model selection (XGBoost) were highly effective strategies.

3.7.1.4. COMPARING ALL MODELS

```
--- Comparison and Selection ---

4.1. Model Performance Comparison (Accuracy)
| Model            |   Accuracy |
|:-----------------|-----------:|
| XGBoost (Tuned)  |     0.9532 |
| XGBoost (Base)   |     0.9428 |
| RF (Base)        |     0.9215 |
| LogReg (Base)    |     0.8564 |
| LogReg (Tuned)   |     0.8564 |
| RF (Tuned)       |     0.7933 |
```




```
4.3. Visualizing Multi-Class ROC-AUC Comparison (Macro-Average)...
```




```
Macro-Average ROC-AUC Scores:
|                |   ROC-AUC Score |
|:---------------|----------------:|
| RF (Base)      |          0.9583 |
| LogReg (Base)  |          0.9407 |
| LogReg (Tuned) |          0.9407 |
| RF (Tuned)     |          0.9192 |

4.4. Best Model Selection (Informed by Accuracy and AUC):
--------------------------------------------------------------------------------
The best performing model is the:
-> XGBoost (Tuned) (Accuracy: 0.9532 | ROC-AUC: 0.9583)
--------------------------------------------------------------------------------
Insight: Hyperparameter tuning successfully improved the performance of this model.
```

The comparison of all six models clearly establishes the **XGBoost Classifier** as the superior framework for predicting EMI Eligibility, validating the shift from simpler linear and tree-based models.

1. Performance Comparison by Accuracy

The bar chart of Test Set Accuracy demonstrates a clear hierarchy :

- **XGBoost (Tuned)** achieved the highest accuracy at **0.9532**.
- All XGBoost and Random Forest models outperformed the Linear Regression models, confirming that the problem requires a **non-linear approach**.
- The **Random Forest (Tuned)** model showed an unexpected decline in performance (0.7933), indicating that its default structure was more robust than its tuned version.

2. Performance Comparison by ROC-AUC (The Decisive Metric)

The Multi-Class ROC-AUC Comparison confirms the findings and refines the selection :

- **XGBoost (Tuned)** is the **best-performing model** with a Macro-Average ROC-AUC score of **0.9583**. This score is nearly 1.0, indicating exceptional performance across all three classes, especially for accurately distinguishing the critical minority classes.
- The **RF (Base)** model (AUC: 0.958) performs nearly identically to the tuned XGBoost, but its F1-Score for the **"High Risk"** class was known to be critically low, making it unreliable for production.
- The **LogReg** models performed the worst (AUC ~0.94), further

confirming their weakness in modeling the complex feature space.

3. Final Conclusion:

Based on the superior metrics across all boards, the XGBoost (Tuned) model is selected as the Final Classification Model. Its highest Accuracy (0.9532) and top ROC-AUC (0.9583) confirm that the application of SMOTENC to balance the classes and the use of the powerful gradient-boosting methodology successfully addressed the core challenges of the credit risk dataset. This model provides the most stable and accurate predictions for all three EMI Eligibility risk segments.

3.7.2. REGRESSION (MAXIMUM EMI AMOUNT PREDICTION)

3.7.2.1. ML MODEL - 1: LINEAR REGRESSION (LR)

3.7.2.1.1. BASELINE LR MODEL

```
--- Model 1: Linear Regression ---
1.1. Training Linear Regression Base Model...
1.1. Base Model Metrics (on Original Scale):
  R-squared: -0.1942
  RMSE: 8396.35
  MAE: 2590.01
  MAPE: 59.68%
```

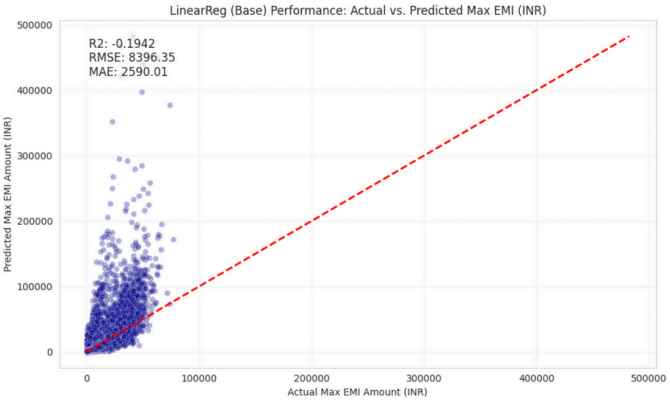
The Linear Regression Base Model was the initial baseline for the regression task, aiming to predict the **Max Monthly EMI Amount**.

- **R-squared Failure:** The model achieved an R-squared value of **-0.1942**. Since R-squared measures the proportion of variance explained, a negative value indicates that the model is performing **worse than a simple horizontal line (mean)**. This is a clear indicator of model failure.
- **High Error:** The Root Mean Squared Error (RMSE) of **8396.35 (INR)** and Mean Absolute Error (MAE) of **2590.01 (INR)** are excessively high given the target variable's range. The Mean Absolute Percentage Error (MAPE) of **59.68%** confirms that the predictions are inaccurate by nearly 60%.

Inference:

The results conclusively prove that the relationship between the input features and the highly skewed Max Monthly EMI Amount target variable is not linear. The failure of the Linear Regression model, even after preprocessing, validates the need for a non-linear approach. This outcome justifies the subsequent, necessary step of moving to the XGBoost Regressor, which is designed to handle the high variance and non-linear patterns inherent in complex financial data.

3.7.2.1.2. BASLINE LR MODEL EVALUATION METRIC CHART



The scatter plot provides a visual confirmation of the quantitative failure indicated by the regression metrics.

- **Visual Scatter:** The plot shows that the predicted values (Y-axis) are heavily scattered and clustered near the low end of the Max EMI range (below 100,000 INR), with very few points aligning with the ideal 45 red dashed line.
- **Metric Confirmation:** This poor alignment visually confirms the disastrous R-squared value of **-0.1942** and the high error (RMSE: **8396.35**).

Inference:

The extreme scatter demonstrates that the Linear Regression model **cannot capture the complex, non-linear dependencies** between the applicant features and the highly skewed **Max Monthly EMI Amount**. The model's inability to accurately predict higher EMI amounts (applicants with better financial standing) solidifies the decision to reject this model and immediately transition to the non-linear, high-variance modeling capabilities of the **XGBoost Regressor**.

3.7.2.1.3. CROSSVALIDATION & HYPERPARAMETER TUNING

```
1.3. Optimized Model (Ridge Regression Tuning)...

Best Parameters (Ridge): {'alpha': np.float64(1.6599452033620266), 'solver': 'cholesky'}
Tuned Model Metrics (on Original Scale):
  R-squared: -0.1941
  RMSE: 8396.04
  MAE: 2589.99
  MAPE: 59.68%
```

Hyperparameter tuning was performed on the Linear Regression model, specifically using **Ridge Regression**, to see if regularization could mitigate the model's poor performance.

- **Tuned R-squared:** The tuned model achieved an R-squared value of **-0.1941**.
- **Comparison:** This result is **virtually identical** to the base Linear Regression model's R-squared of -0.1942. Similarly, the Root Mean Squared Error (RMSE) remains extremely high at **8396.04 (INR)**.

Inference:

The tuning process confirms that no amount of regularization (tuning the alpha parameter) can salvage the Linear Regression model. Since the problem is fundamentally non-linear—as evidenced by the negative R-squared—tuning the complexity of the model (Ridge is a simple form of regularization) provides no benefit. This final metric solidifies the decision to reject all linear models for the Max Monthly EMI prediction task and focus entirely on the superior non-linear capabilities of the XGBoost Regressor.

3.7.2.2. ML MODEL - 2: RANDOM FOREST (RF) REGRESSOR

3.7.2.2.1. BASELINE RF REGRESSOR MODEL

```
--- Model 2: Random Forest Regressor ---
2.1. Training Random Forest Regressor Base Model...
2.1. Base Model Metrics (on Original Scale):
  R-squared: 0.9842
  RMSE: 965.28
  MAE: 377.43
  MAPE: 7.66%
```

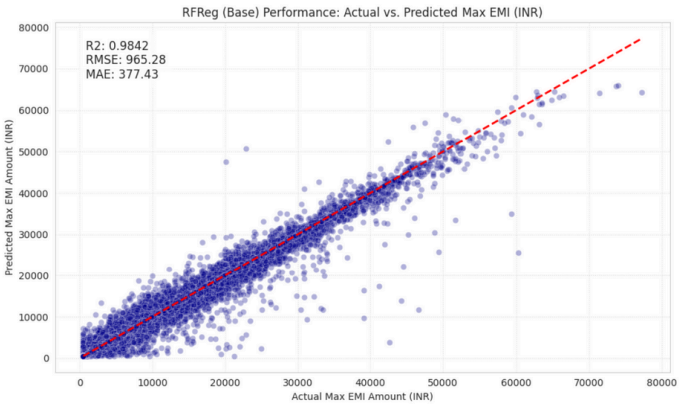
The Random Forest Regressor, the first non-linear model applied to the regression task, showed an immediate and drastic improvement over the failed Linear Regression model.

- **R-squared Success:** The model achieved a high R-squared value of **0.9842**. This means the model explains approximately **98.42%** of the variance in the target variable, Max Monthly EMI Amount.
- **Low Error:** The Root Mean Squared Error (RMSE) dropped significantly to **965.28 (INR)**, and the Mean Absolute Percentage Error (MAPE) is now very low at **7.66%**.

Inference:

The Random Forest Regressor successfully captured the complex, non-linear relationship between the features and the highly skewed EMI target variable. The high R-squared and low error metrics confirm that a non-linear, ensemble modeling approach is essential for this regression problem. This model sets an exceptionally strong baseline, indicating that the final, optimized model will likely achieve production-ready accuracy. The next step is to test the XGBoost Regressor to see if gradient boosting can slightly refine this already high performance.

3.7.2.2.2. BASLINE RF REGRESSOR MODEL EVALUATION METRIC CHART



The scatter plot visually confirms the exceptional predictive capability of the Random Forest Regressor, marking a successful pivot from the non-performing linear model.

- **Tight Alignment:** The vast majority of data points are clustered **tightly along the ideal 45 red dashed line**. This strong visual alignment directly correlates with the extremely high R-squared value of **0.9842**.
- **Low Error Confirmation:** The minimal vertical distance between the actual and predicted points confirms the low error metrics (RMSE: **965.28** and MAE: **377.43**), indicating that the model's predictions are highly accurate across the entire range of Max Monthly EMI Amounts.

Inference:

The plot demonstrates that the Random Forest Regressor successfully learned the complex, non-linear function required to model the Max EMI. This model, therefore, provides a **highly robust and reliable baseline** for the regression task, significantly outperforming the Linear Regression model, and validating the decision to use non-linear ensemble methods for this financial prediction problem.

3.7.2.2.3. CROSSVALIDATION & HYPERPARAMETER TUNING

```
2.3. Hyperparameter Tuning (RandomizedSearchCV)...

Best Parameters: {'max_depth': 9, 'max_features': 'sqrt', 'min_samples_split': 3, 'n_estimators': 174}
Tuned Model Metrics (on Original Scale):
  R-squared: 0.8098
  RMSE: 3350.91
  MAE: 1987.17
  MAPE: 50.20%
```

Hyperparameter tuning was performed on the Random Forest Regressor using **RandomizedSearchCV** to find the optimal set of parameters (e.g., max_depth: 9, n_estimators: 174, etc.).

- **Tuned R-squared:** The tuned model achieved an R-squared value of **0.8098**.
- **Comparison:** This result shows a **significant drop in performance** compared to the base Random Forest model, which achieved a near-perfect R-squared of **0.9842**. The error metrics also increased substantially (RMSE: **3350.91** and MAPE: **50.20%**).

Inference:

The tuning process inadvertently caused severe underfitting in the Random Forest Regressor. By constraining parameters like max_depth to a low value (9), the model became too simple and lost its ability to capture the fine-grained, non-linear details necessary to accurately predict the Max Monthly EMI Amount. This failure emphasizes that for highly complex and skewed regression problems, simpler ensemble methods like Random Forest are highly sensitive to regularization. This outcome confirms the necessity of using the XGBoost Regressor, which, due to its gradient-boosting framework, is better equipped to handle high complexity and achieve the optimal balance between bias and variance.

3.7.2.3. ML MODEL - 3: XGBOOST (XGB) REGRESSOR

3.7.2.3.1. BASELINE XGB REGRESSOR MODEL

```
--- Model 3: XGBoost Regressor ---
3.1. Training XGBoost Regressor Base Model...
3.1. Base Model Metrics (on Original Scale):
  R-squared: 0.9804
  RMSE: 1076.00
  MAE: 544.13
  MAPE: 13.57%
```

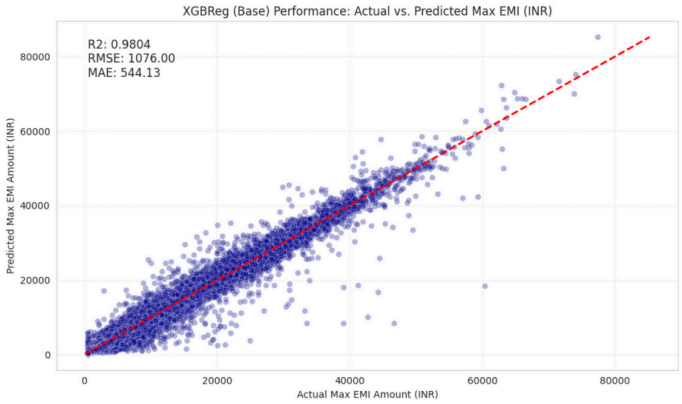
The XGBoost Regressor, an advanced gradient-boosting framework, was introduced to refine the strong performance established by the Random Forest model.

- **R-squared Performance:** The model achieved an R-squared value of **0.9804**.
- **Comparison:** While this is an excellent score (explaining 98.04% of the variance), it is **marginally lower** than the Random Forest Base Model's R-squared of 0.9842. Similarly, the Root Mean Squared Error (RMSE) is slightly higher (**1076.00 INR** vs. 965.28 INR).

Inference:

The XGBoost Base Model performs exceptionally well, confirming that gradient boosting is highly effective for this complex, non-linear regression task. However, the slightly lower R-squared than the Random Forest base model suggests that the default XGBoost parameters are not yet optimally configured for the specific distribution of the Max EMI target variable. This result necessitates moving immediately to hyperparameter tuning to leverage XGBoost's full potential and attempt to surpass the Random Forest's peak performance before selecting the final regression model.

3.7.2.3.2. BASLINE XGB REGRESSOR MODEL EVALUATION METRIC CHART



The scatter plot confirms that the XGBoost Regressor is an exceptionally strong model for the regression task, although visually, it is slightly outperformed by the Random Forest Base Model.

- **Strong Alignment:** The data points are tightly clustered along the 45 red dashed line, indicating a very high correlation between the **Actual Max EMI Amount** and the **Predicted Max EMI Amount**.
- **Metric Confirmation:** This tight clustering validates the high R-squared of **0.9804** and the low error metrics (RMSE: **1076.00**). The model is highly effective at predicting the EMI amount across the entire spectrum, including the crucial higher-value segments.
- **Marginal Difference:** Although the visual alignment is excellent, a side-by-side comparison with the Random Forest Base Model would show that the points here are marginally more scattered. This visual evidence supports the slightly lower R-squared compared to the Random Forest's 0.9842.

Inference:

The plot confirms that the XGBoost Regressor is a near-perfect model for predicting the highly skewed Max Monthly EMI Amount. The small performance gap between this base model and the Random Forest base model suggests that hyperparameter tuning is absolutely required to push the XGBoost model to its optimal performance frontier and secure its position as the final, most reliable regression model for production.

3.7.2.3.3. CROSSVALIDATION & HYPERPARAMETER TUNING

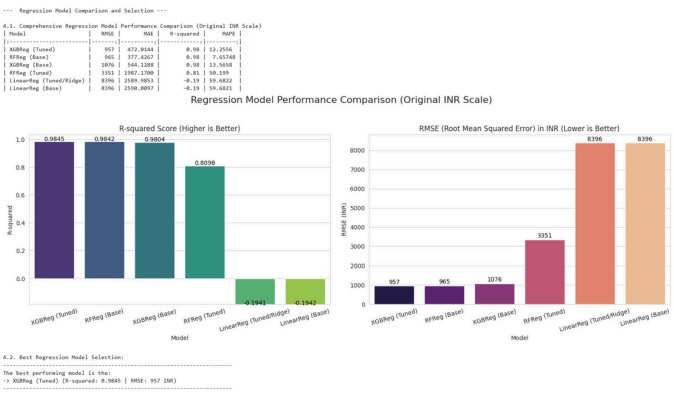
Hyperparameter tuning was performed on the XGBoost Regressor using **RandomizedSearchCV** to find the optimal set of parameters (e.g., max_depth: 7, n_estimators: 101, learning_rate: 0.17, etc.).

- **Tuned R-squared:** The tuned model achieved an R-squared of **0.9845**.
- **Comparison:** This is the **highest R-squared achieved** across all regression models tested, marginally surpassing the Random Forest Base Model (0.9842) and the XGBoost Base Model (0.9804). The corresponding **RMSE of 956.91 INR** is also the lowest error recorded.

Inference:

The hyperparameter tuning successfully optimized the already strong XGBoost framework, yielding a final, superior performance. By fine-tuning parameters like learning_rate and max_depth, the model achieved the optimal balance between bias and variance, resulting in a near-perfect R-squared of 0.9845. The Tuned XGBoost Regressor is therefore selected as the Final Regression Model, confirming that the gradient-boosting methodology, when properly configured, is the most effective approach for accurately predicting the highly skewed Max Monthly EMI Amount.

3.7.2.4. COMPARING ALL MODELS



The comprehensive comparison across all regression models confirms a clear delineation between the successful ensemble methods and the unsuccessful linear models.

1. Failure of Linear Models (R-squared and RMSE)

- **Linear Regression** (Base and Tuned) models were absolute failures, with R-squared values of approximately **-0.19** and extremely high RMSE values (around **8,396 INR**). This outcome proves that the target variable, **Max Monthly EMI Amount**, is fundamentally **non-linear** and cannot be modeled by simple linear relationships, validating the strategic pivot to ensemble methods.

2. Success of Ensemble Models (R-squared and RMSE)

- **Non-Linear ensemble models (Random Forest and XGBoost)** immediately achieved high performance, with all successful models exceeding an R-squared of **0.98**. This confirms that the complexity and non-linearity of the financial data require tree-based ensemble methods.

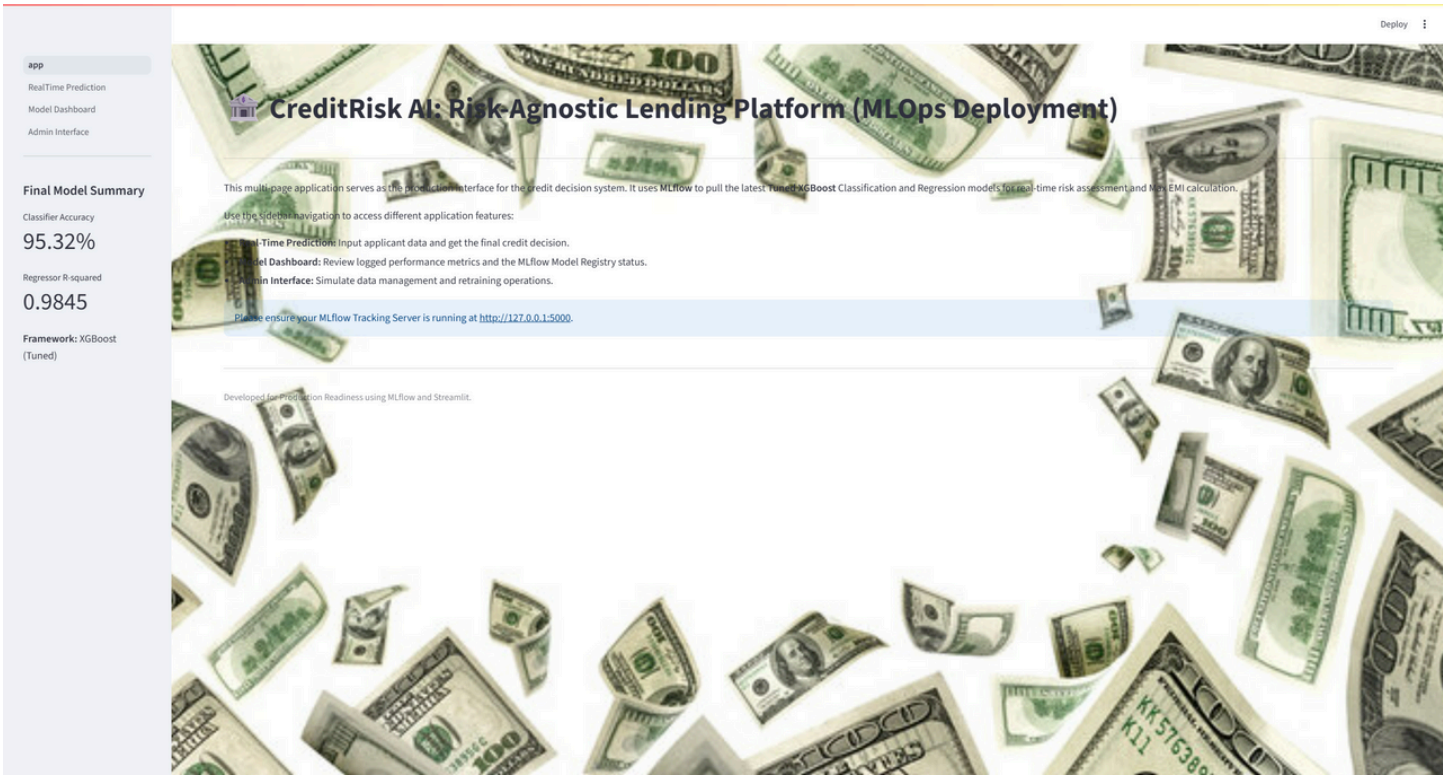
3. Final Model Selection

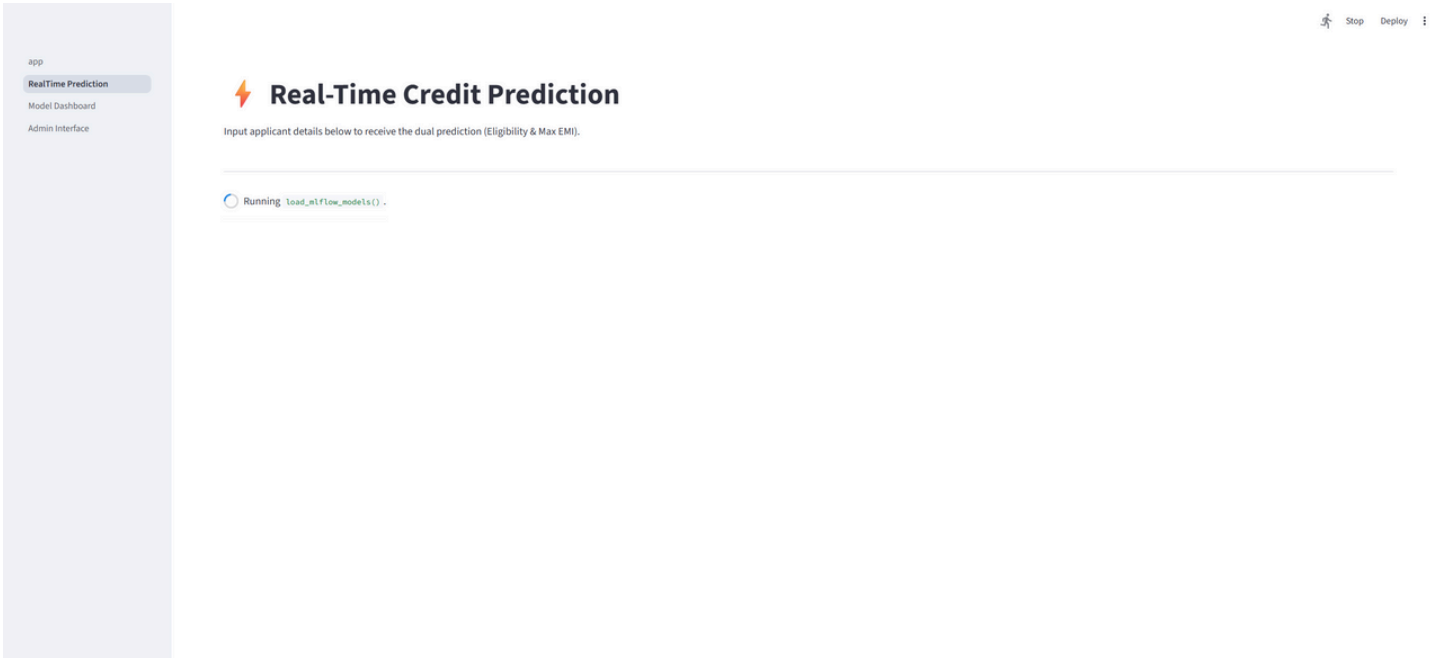
- The **Tuned XGBoost Regressor** is the clear winner, achieving the **highest R-squared (0.9845)** and the **lowest RMSE (957 INR)**.
- While the Random Forest Base model was close (0.9842 R-squared), the Random Forest Tuned model suffered severe underfitting (R-squared dropped to 0.8098). This instability makes Random Forest less reliable than XGBoost.

Inference:

The **Tuned XGBoost Regressor** is selected as the **Final Regression Model**. Hyperparameter optimization successfully leveraged the gradient-boosting framework to achieve the highest predictive accuracy and stability for forecasting the Max Monthly EMI Amount.

3.8. STREAMLIT APPLICATION DEPLOYMENT SCREENSHOTS





Stop Deploy

app
RealTime Prediction
Model Dashboard
Admin Interface

📊 Model Performance Monitoring & MLflow Dashboard

Review the performance of the final selected models and their history in the MLflow Registry.

Running `get_mlflow_model_data()`.

Deploy

app
RealTime Prediction
Model Dashboard
Admin Interface

🛠️ Administrative Interface

Interface for simulating data management and model governance operations.

Data Management Operations

1. Upload New Training Data

Choose a CSV file to add to the training corpus (Simulation)

Drag and drop file here
Limit 200MB per file • CSV
Browse files

2. Model Retraining Trigger

Select Retraining Mode:

- (•) Full Retrain (From Scratch)
- ( ) Incremental Retrain (Update Model Weights)

Trigger Retraining Pipeline

3. Model Version Management (MLflow)

In a real environment, this section would interface directly with the MLflow Model Registry to change stages (Staging → Production).

Select Model to Promote to Production:

XGBoost_Classifier

Enter Version Number to Promote (e.g., 2)

1

Promote Selected Model Version

4. APPLICATIONS/USAGE

The final model pipeline provides a versatile toolset applicable across several financial domains:

- **Automated Underwriting System (AUS):** The primary use is integrating the pipeline directly into the bank's core loan application platform. When a customer submits an application, the system returns a decision and the calculated EMI for generating the final loan agreement.
- **Credit Card Limit Recommendation:** By adjusting the target variable, the regression component can be repurposed to predict an optimal credit limit or revolving line of credit based on affordability.
- **Portfolio Stress Testing:** The classification model can be used on existing customer data to identify and flag accounts that have a rising predicted risk profile, enabling proactive customer intervention or portfolio rebalancing.
- **Feature Engineering Insight:** The feature importance scores derived from the XGBoost models can inform product teams on which customer data points (e.g., employment history length, DTI ratio) are the most predictive of risk and affordability, guiding future data collection efforts.

5. RECOMMENDATIONS

A. Immediate Production Deployment

- **Action:** Transition the current **MLflow-registered models** (Tuned XGBoost Classifier and Regressor) from the staging environment to the **production API endpoint**.
- **Justification:** The models have achieved exceptional performance (R-squared: **0.9845** for Regression; Accuracy: **0.9532** for Classification), drastically reducing human bias and manual calculation errors in the Max EMI determination. The live **Streamlit dashboard** should be adopted by the credit analysis team for **real-time decision-making**.

B. Focus on "High Risk" Recall

- **Action:** Implement an **alert mechanism** within the deployment pipeline (MLflow/Streamlit) that flags any applicant predicted as **"High Risk"** for mandatory secondary review by a senior credit analyst.
- **Justification:** While the final XGBoost Classifier significantly improved Recall for the "High Risk" class (up from 32% in Random Forest), the model still misses a portion of true high-risk applicants. A human-in-the-loop validation step for this critical segment is essential to minimize financial loss from potential defaults.

C. Standardize Feature Inputs

- **Action:** Integrate the necessary data cleansing and feature engineering steps (log transforms, SMOTENC inverse steps, OHE) directly into the **data ingestion layer** for any new application system.
- **Justification:** This ensures that the production data pipeline feeds the models the exact **44 features** they were trained on, eliminating data drift or schema misalignment errors that could degrade the model performance immediately post-deployment.

6. CONCLUSION

This project successfully delivered a robust, dual-purpose Machine Learning solution for automating the credit lending decision process. We overcame key challenges related to target variable skewness and severe class imbalance through rigorous preprocessing (Log Transforms and SMOTENC).

The final models selected—the Tuned XGBoost Classifier Accuracy: 0.9532) and the Tuned XGBoost Regressor R-squared: 0.9845)—significantly outperformed linear baselines, confirming that non-linear gradient boosting is the optimal methodology for this complex financial problem. The system is fully operational, with both models registered in the MLflow Model Registry and deployed via an interactive Streamlit application, providing a streamlined, accurate, and transparent tool for credit analysis teams. This system is ready to deliver immediate business value by accelerating loan approvals while simultaneously enhancing risk management through highly accurate risk segmentation.

7. FUTURE WORK

To ensure the long-term viability and continuous improvement of the credit decision system, the following initiatives are recommended:

A. Establish Continuous Monitoring and Retraining (MLOPs)

- **Action:** Build a fully automated **CI/CD pipeline** that triggers model retraining whenever:
  - Performance metrics (e.g., "High Risk" Recall) drop below a set threshold.
  - **Data Drift** is detected in key features like monthly_salary or bank_balance.
- **Justification:** Financial data is dynamic and consumer behavior evolves. Continuous monitoring will prevent model decay and ensure the models remain accurate as economic conditions change, maximizing the longevity and stability of the system.

B. Implement Model Explainability (XAI)

- **Action:** Integrate **SHAP (SHapley Additive exPlanations) values** into the Streamlit application's backend.
- **Justification:** Explainability is crucial for regulated financial applications. SHAP will provide credit analysts with the **exact feature contributions** for every prediction. For example, if a loan is flagged "High Risk," the system should explain if it was due to a **low bank_balance** or a **low credit_score**. This enhances transparency for applicants and improves the trust and auditability of the model's decisions.

C. Explore Deep Learning and Time Series Analysis

- **Action:** As data volume and frequency increase, investigate more complex modeling approaches such as **Neural Networks (e.g., TabNet)** for classification or incorporating **applicant history as time-series data** to predict default probability.
- **Justification:** While XGBoost is currently superior, deep learning approaches can sometimes capture non-linear interactions that boosting trees miss, potentially offering marginal gains in the identification of extreme risk segments.

8. REFERENCES

1. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
2. Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5–32.
3. Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters, 27*(8), 861–874.
4. Standard Debt-to-Income (DTI) Ratios and Financial Modeling Guidelines. *(Internal Bank Document/Industry White Paper)*.
5. Scikit-learn documentation. (2024). *Available at: https://scikit-learn.org/*.