Rakamin Academy X id/x partners

Prediction Modeling

Credit Risk Analysis

ID/X Partners - Data Scientist

Presented by Ahmad Fauzi

ahmadfaauzi2304@gmail.com

github.com/ahmfzui

https://www.linkedin.com/in/ahmad-fauzi-03504320b/

Rakamin Academy

AHMAD FAUZI

- **Education:** Undergraduate Degree in Information Systems at Telkom University
- **Experience:**
  - Member of Study Group at Enterprise Data Management (EDM) Lab
  - Former Teaching Assistant for Algorithm Programming Practicum

My keen interest lies in the transformative power of data, driving me to explore its depth and uncover valuable insights that empower informed decisions and innovation.

Project Portfolio

As an intern, I am involved in developing a machine learning model using loan data from Lending Club (2007-2014) to predict credit risk, aimed at enhancing business decision-making accuracy and minimizing losses. Our focus includes key business metrics such as loss mitigation and net profit margins. This data analysis seeks to uncover patterns indicative of potentially risky or poor-performing loans, facilitating informed investment decisions without making strong assumptions.

Project explanation video here!

About Company

ID/X Partners was founded in 2002 by former bankers and management consultants with extensive experience in credit lifecycle management, scoring development, and performance management. Our collective expertise has served corporations across Asia, Australia, and various industries, including financial services, telecommunications, manufacturing, and retail.

ID/X Partners specializes in leveraging data analytics and decision-making solutions (DAD), integrated with risk management and marketing disciplines, to optimize portfolio profitability and business processes for our clients.

Our comprehensive consulting services and technology solutions make ID/X Partners a leading integrated service provider in the industry.

Business & Data Understanding

Rakamin Academy

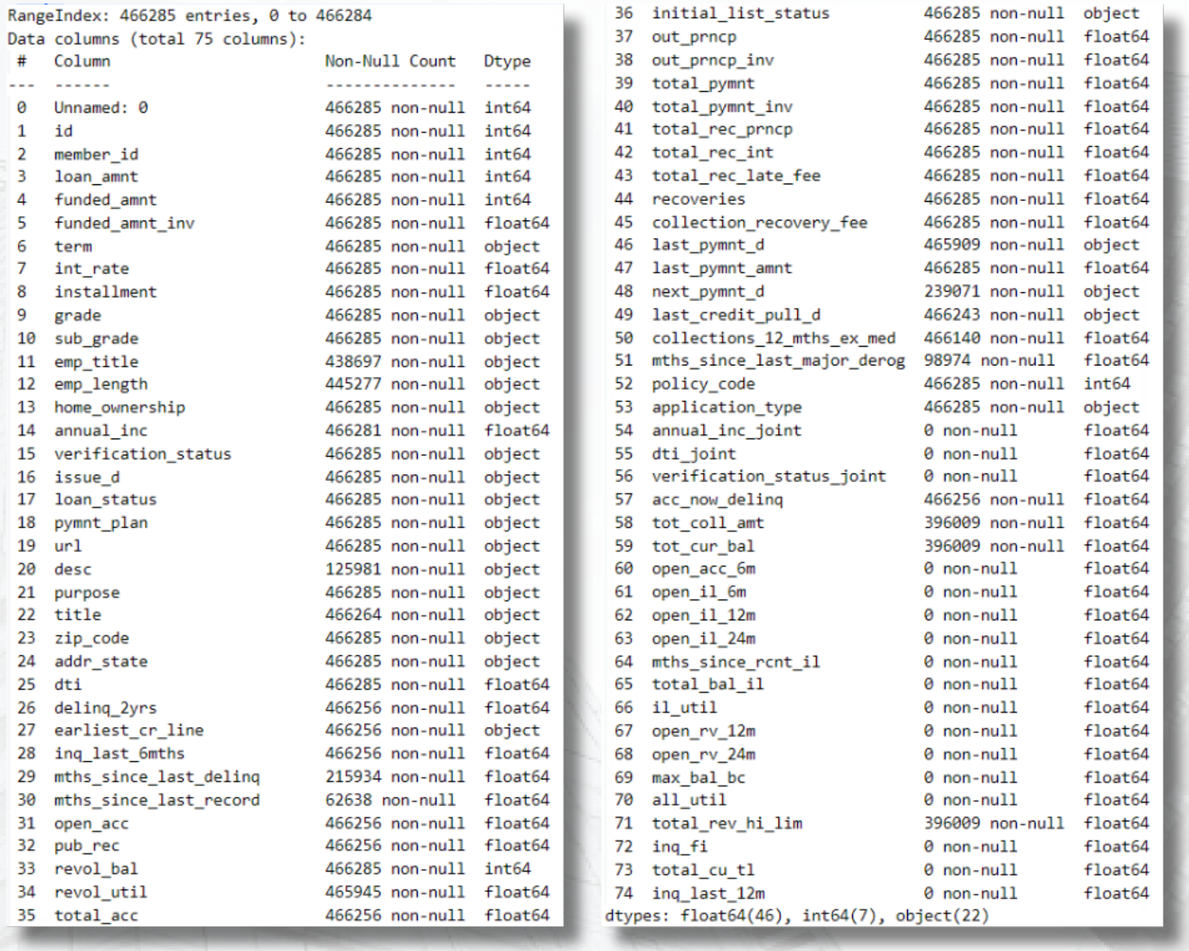
Business Understanding

- **Credit Risk Overview:** Credit risk refers to the potential loss associated with the failure to meet loan repayment obligations at maturity. It is a critical factor in the lending process for financial institutions.
- **Importance of Credit Risk Management:** Effective credit risk management allows financial institutions to make informed lending decisions, optimizing their portfolio profitability while minimizing the risk of borrower default.
- **Project Objective:** The objective of this project is to enhance credit risk assessment accuracy for a multifinance company. By leveraging historical borrower data and machine learning algorithms, we aim to develop a predictive model that can effectively forecast credit risk.

Data Understanding

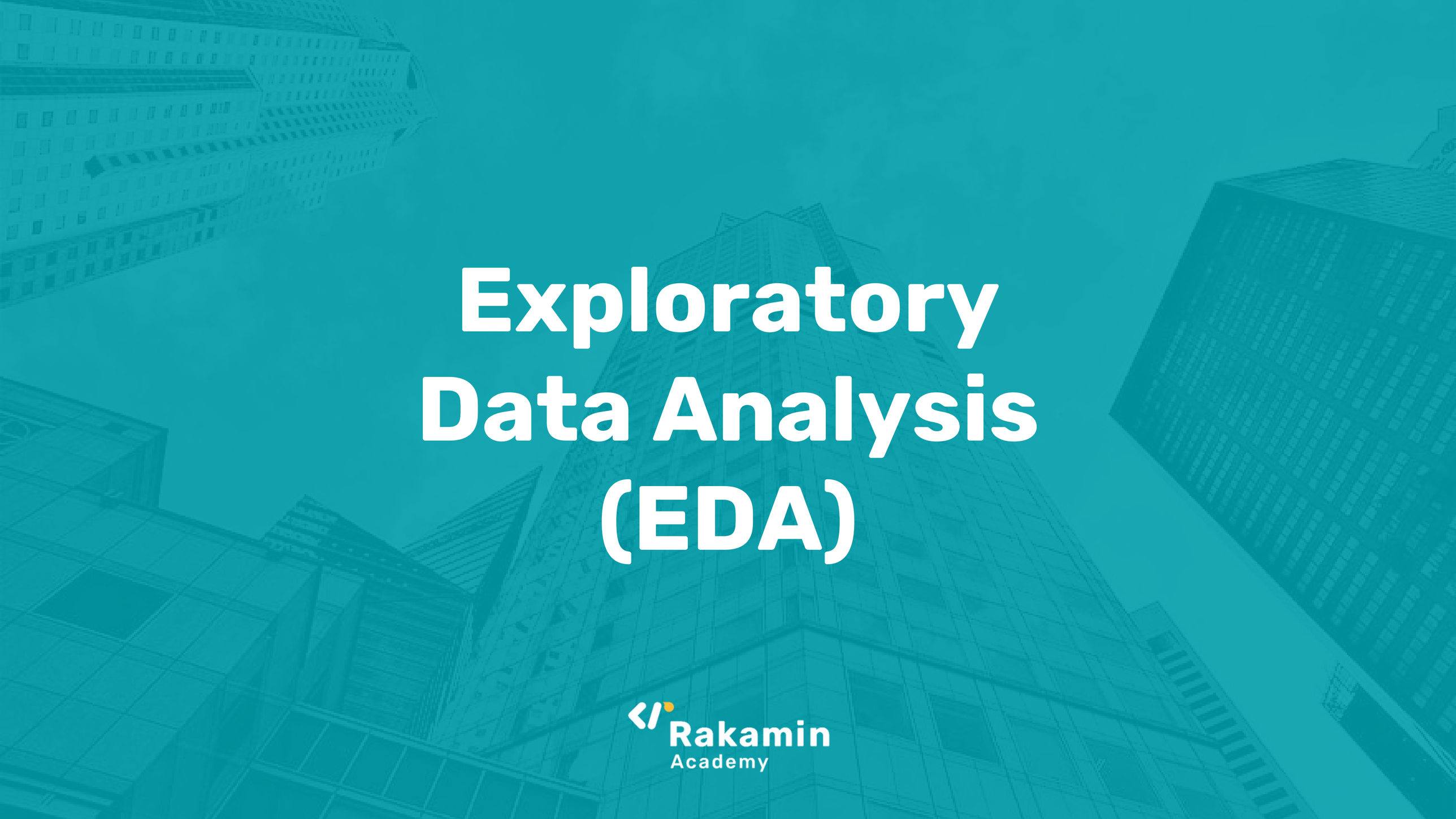
Rakamin Academy

- **Dataset Source:** The dataset provided by the company originates from loan records gathered over several years.
- **Number of Attributes:** There are 75 attributes (columns) in the dataset.
- **Number of Rows:** The dataset contains 466,285 records (rows), indexed by an additional column named "Unnamed: 0".
- **Additional Info:**
  - Dataset consists of 3 data types: int64, float64, and object.
  - The dataset doesn't initially include a target variable (loan_status); **it needs to be created**.
  - Features like issue_d, last_pymnt_d, next_pymnt_d, last_credit_pull_d, and earliest_cr_line should be converted to datetime data types.
  - Forty columns have null values that require handling during data preparation.



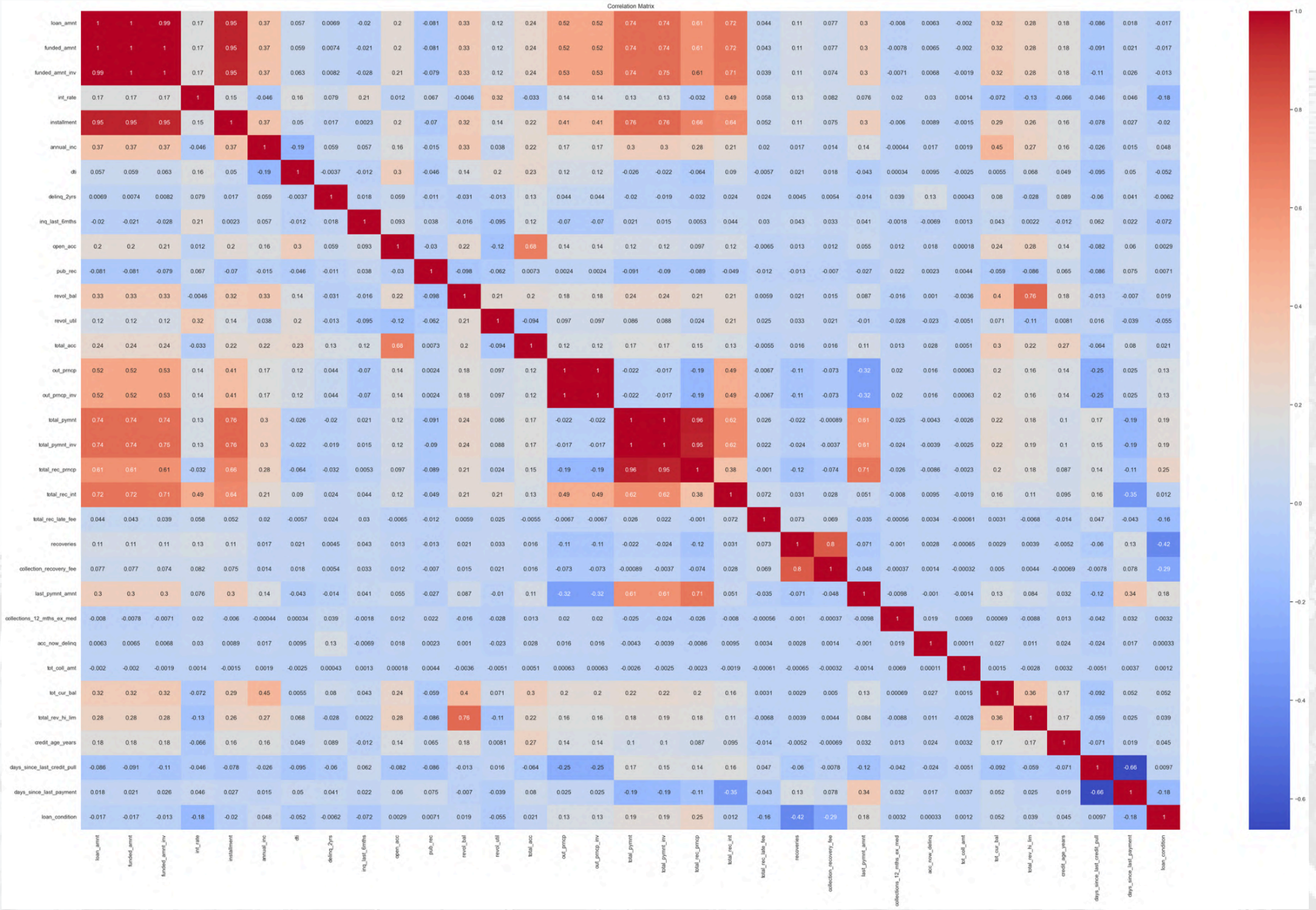
```
RangeIndex: 466285 entries, 0 to 466284
Data columns (total 75 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   Unnamed: 0                  466285 non-null  int64
 1   id                          466285 non-null  int64
 2   member_id                   466285 non-null  int64
 3   loan_amnt                   466285 non-null  int64
 4   funded_amnt                 466285 non-null  int64
 5   funded_amnt_inv             466285 non-null  float64
 6   term                        466285 non-null  object
 7   int_rate                    466285 non-null  float64
 8   installment                 466285 non-null  float64
 9   grade                       466285 non-null  object
 10  sub_grade                   466285 non-null  object
 11  emp_title                   438697 non-null  object
 12  emp_length                  445277 non-null  object
 13  home_ownership              466285 non-null  object
 14  annual_inc                  466281 non-null  float64
 15  verification_status         466285 non-null  object
 16  issue_d                     466285 non-null  object
 17  loan_status                 466285 non-null  object
 18  pymnt_plan                  466285 non-null  object
 19  url                         466285 non-null  object
 20  desc                        125981 non-null  object
 21  purpose                     466285 non-null  object
 22  title                       466264 non-null  object
 23  zip_code                    466285 non-null  object
 24  addr_state                  466285 non-null  object
 25  dti                         466285 non-null  float64
 26  delinq_2yrs                 466256 non-null  float64
 27  earliest_cr_line            466256 non-null  object
 28  inq_last_6mths              466256 non-null  float64
 29  mths_since_last_delinq      215934 non-null  float64
 30  mths_since_last_record      62638 non-null   float64
 31  open_acc                    466256 non-null  float64
 32  pub_rec                     466256 non-null  float64
 33  revol_bal                   466285 non-null  int64
 34  revol_util                  465945 non-null  float64
 35  total_acc                   466256 non-null  float64
 36  initial_list_status         466285 non-null  object
 37  out_prncp                   466285 non-null  float64
 38  out_prncp_inv               466285 non-null  float64
 39  total_pymnt                 466285 non-null  float64
 40  total_pymnt_inv             466285 non-null  float64
 41  total_rec_prncp             466285 non-null  float64
 42  total_rec_int               466285 non-null  float64
 43  total_rec_late_fee          466285 non-null  float64
 44  recoveries                  466285 non-null  float64
 45  collection_recovery_fee     466285 non-null  float64
 46  last_pymnt_d                465909 non-null  object
 47  last_pymnt_amnt             466285 non-null  float64
 48  next_pymnt_d                239071 non-null  object
 49  last_credit_pull_d          466243 non-null  object
 50  collections_12_mths_ex_med  466140 non-null  float64
 51  mths_since_last_major_derog 98974 non-null   float64
 52  policy_code                 466285 non-null  int64
 53  application_type            466285 non-null  object
 54  annual_inc_joint            0 non-null       float64
 55  dti_joint                   0 non-null       float64
 56  verification_status_joint   0 non-null       float64
 57  acc_now_delinq              466256 non-null  float64
 58  tot_coll_amt                396009 non-null  float64
 59  tot_cur_bal                 396009 non-null  float64
 60  open_acc_6m                 0 non-null       float64
 61  open_il_6m                  0 non-null       float64
 62  open_il_12m                 0 non-null       float64
 63  open_il_24m                 0 non-null       float64
 64  mths_since_rcnt_il          0 non-null       float64
 65  total_bal_il                0 non-null       float64
 66  il_util                     0 non-null       float64
 67  open_rv_12m                 0 non-null       float64
 68  open_rv_24m                 0 non-null       float64
 69  max_bal_bc                  0 non-null       float64
 70  all_util                    0 non-null       float64
 71  total_rev_hi_lim            396009 non-null  float64
 72  inq_fi                      0 non-null       float64
 73  total_cu_tl                 0 non-null       float64
 74  inq_last_12m                0 non-null       float64
dtypes: float64(46), int64(7), object(22)
```

Exploratory Data Analysis (EDA)



Correlation and Multicollinearity Analysis

	Feature 1	Feature 2	Correlation
0	out_prncp	out_prncp_inv	0.999998
1	out_prncp_inv	out_prncp	0.999998
2	loan_amnt	funded_amnt	0.998548
3	funded_amnt	loan_amnt	0.998548
4	funded_amnt	funded_amnt_inv	0.996125
5	funded_amnt_inv	funded_amnt	0.996125
6	total_pymnt_inv	total_pymnt	0.995862
7	total_pymnt	total_pymnt_inv	0.995862
8	funded_amnt_inv	loan_amnt	0.994347
9	loan_amnt	funded_amnt_inv	0.994347
10	total_pymnt	total_rec_prncp	0.956658
11	total_rec_prncp	total_pymnt	0.956658
12	total_pymnt_inv	total_rec_prncp	0.952158
13	total_rec_prncp	total_pymnt_inv	0.952158
14	installment	funded_amnt	0.951787
15	funded_amnt	installment	0.951787
16	loan_amnt	installment	0.949666
17	installment	loan_amnt	0.949666
18	installment	funded_amnt_inv	0.947387
19	funded_amnt_inv	installment	0.947387
20	recoveries	collection_recovery_fee	0.800666
21	collection_recovery_fee	recoveries	0.800666



Target Variabel Analysis

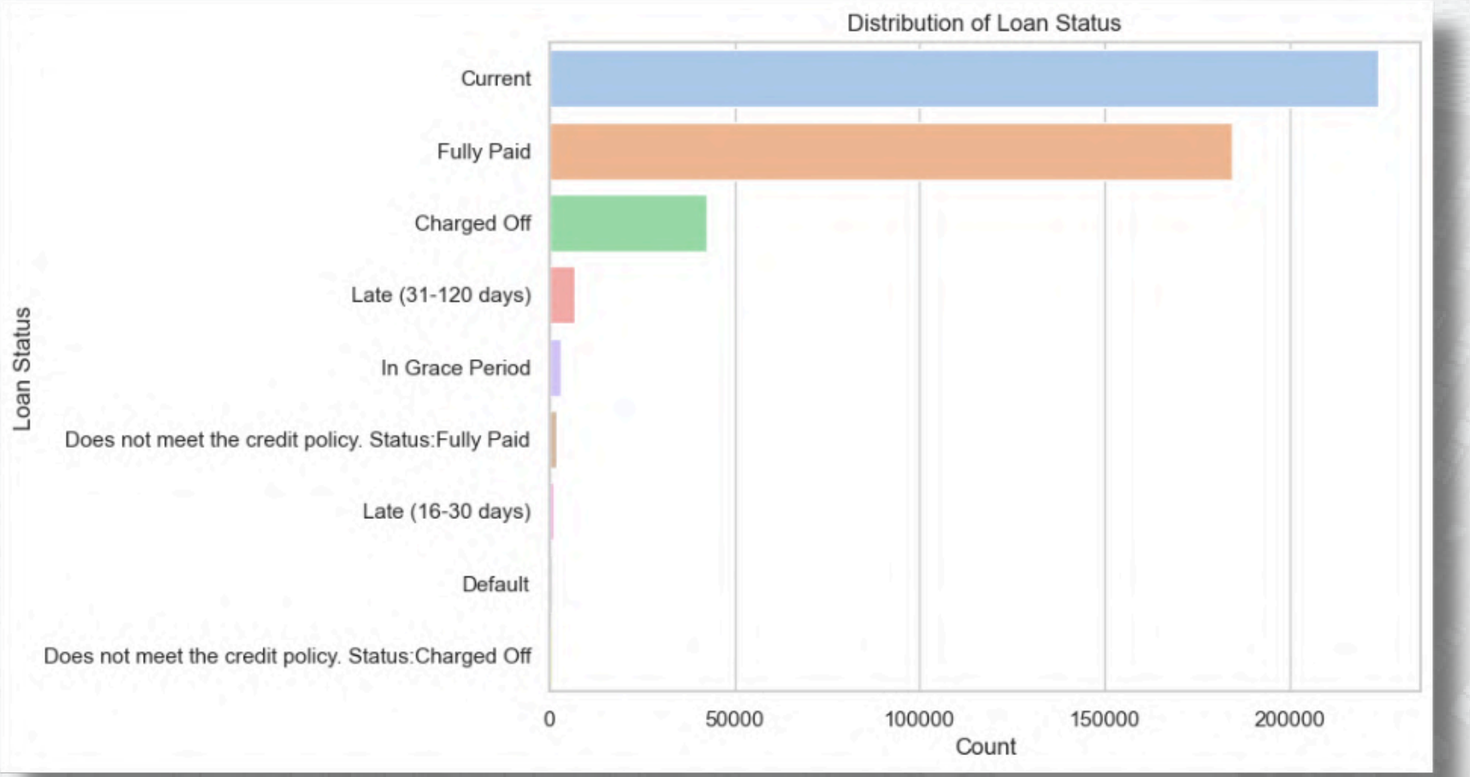
'loan_status' has been mapped to a new column 'loan_condition' based on predefined categories of 'Good' and 'Bad'. This mapping categorizes loans as follows:

Good Loan Status ; 1

- Fully Paid
- Current
- Does not meet the credit policy. Status:Fully Paid

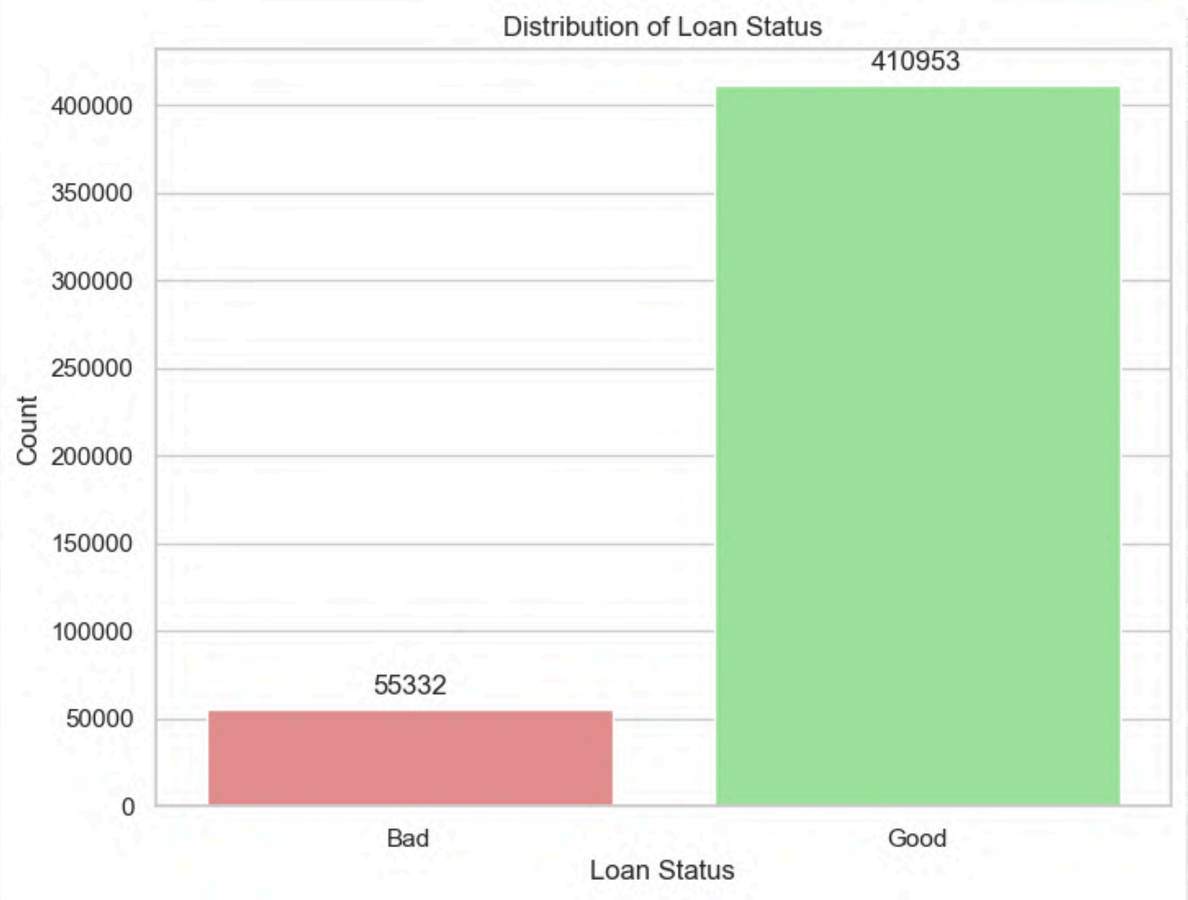
Bad Loan Status ; 0

- Charged Off
- Default
- Late (31-120 days)
- In Grace Period
- Late (16-30 days)
- Does not meet the credit policy. Status:Charged Off



Target Variabel Analysis

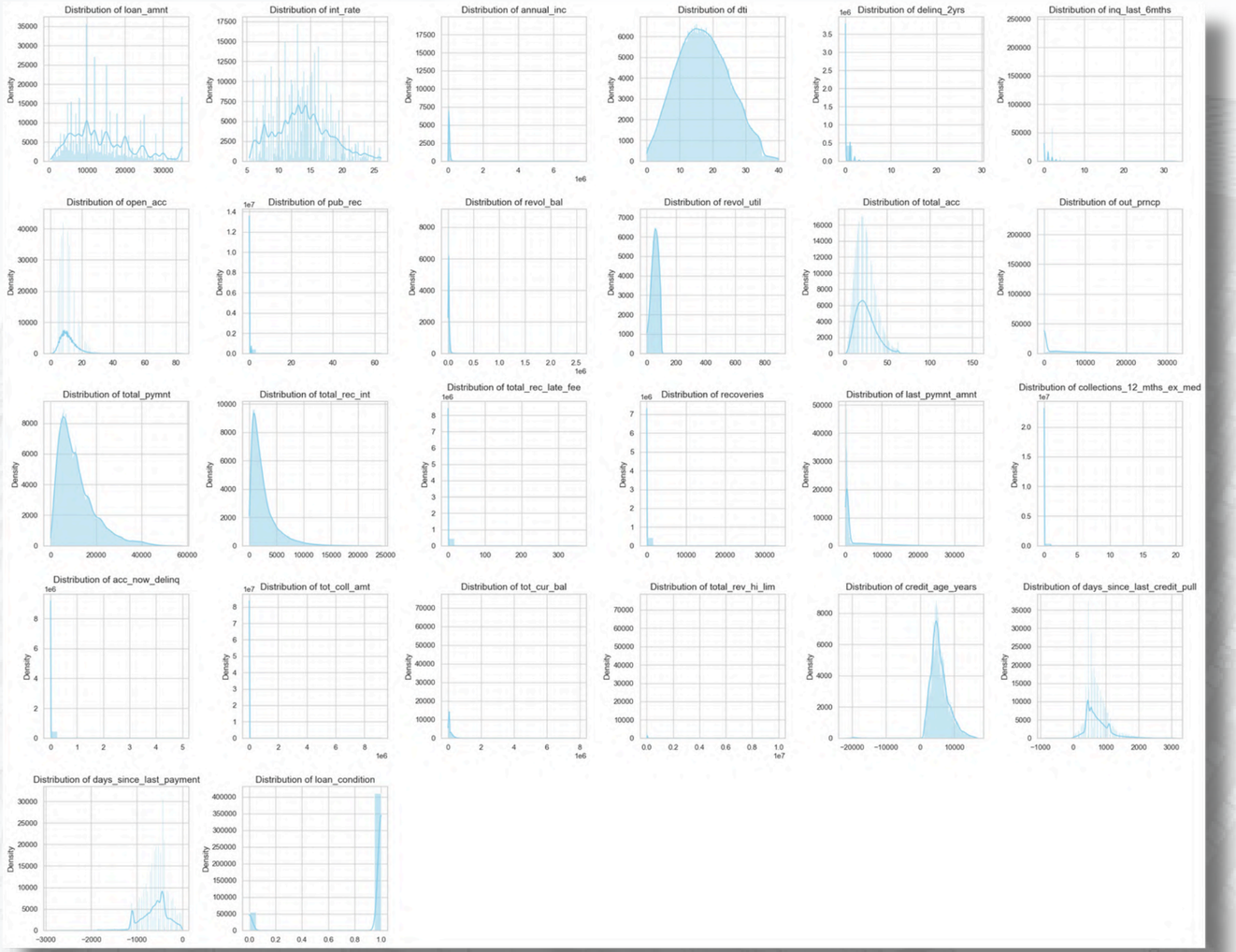
The **dataset imbalance,** where there are significantly more good loans than bad loans, may affect model performance. Consider addressing this with oversampling or undersampling techniques during data preparation.



Distribution and Box Plot Analysis

Observations:

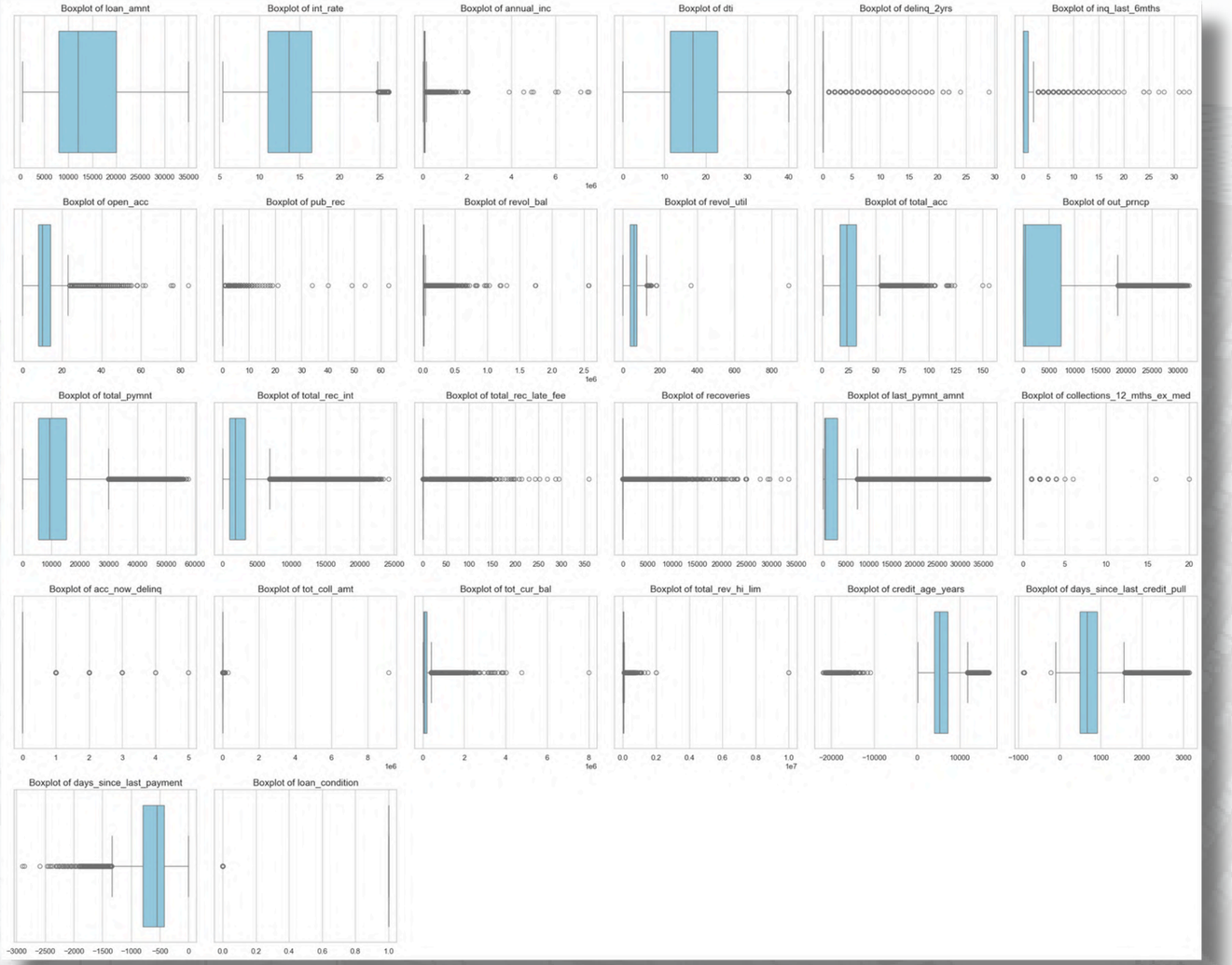
- **Highly Skewed Variables:**
  - annual_inc, revol_bal, total_rec_late_fee, total_rev_hi_lim, and days_since_last_credit_pull
  - These variables show a high concentration of values in a narrow range, with some extremely large or small values.
- **More Normally Distributed Variables:**
  - loan_amnt, int_rate, and dti
  - These variables have a more uniform distribution across their ranges.



Distribution and Box Plot Analysis

Observations:

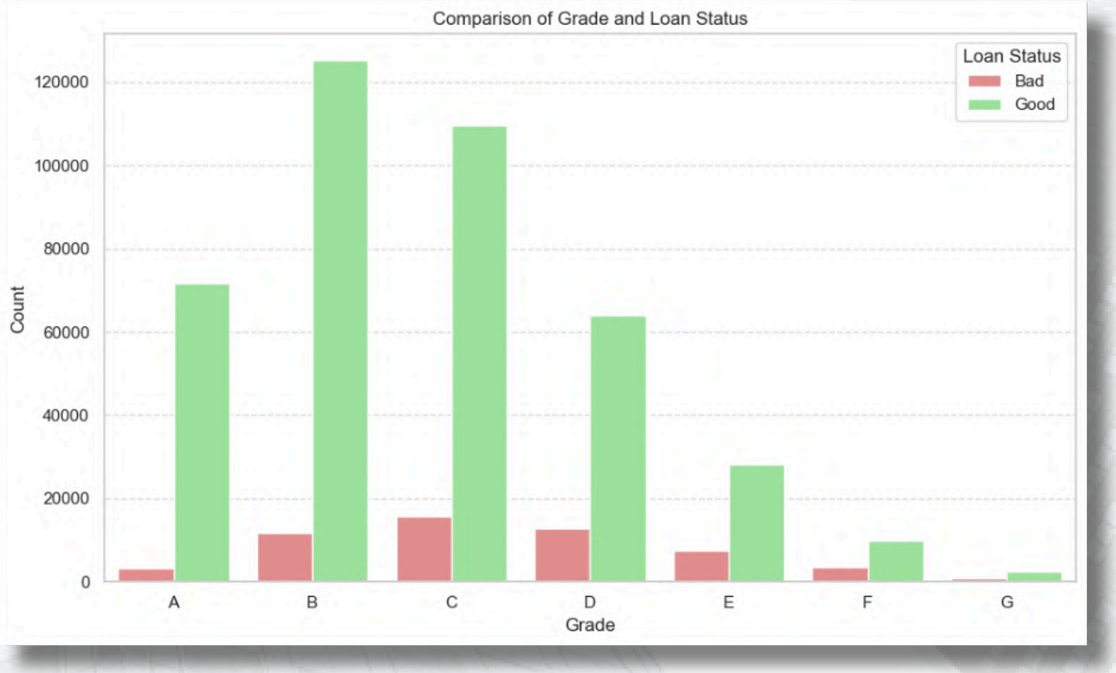
- **Prominent Outliers:**
  - Many variables, such as annual_inc, revol_bal, total_rec_late_fee, total_rev_hi_lim, and days_since_last_credit_pull, exhibit significant outliers.
  - These outliers can distort statistical analysis and predictive models.



Loan Status by Grade Analysis

Grade Analysis:

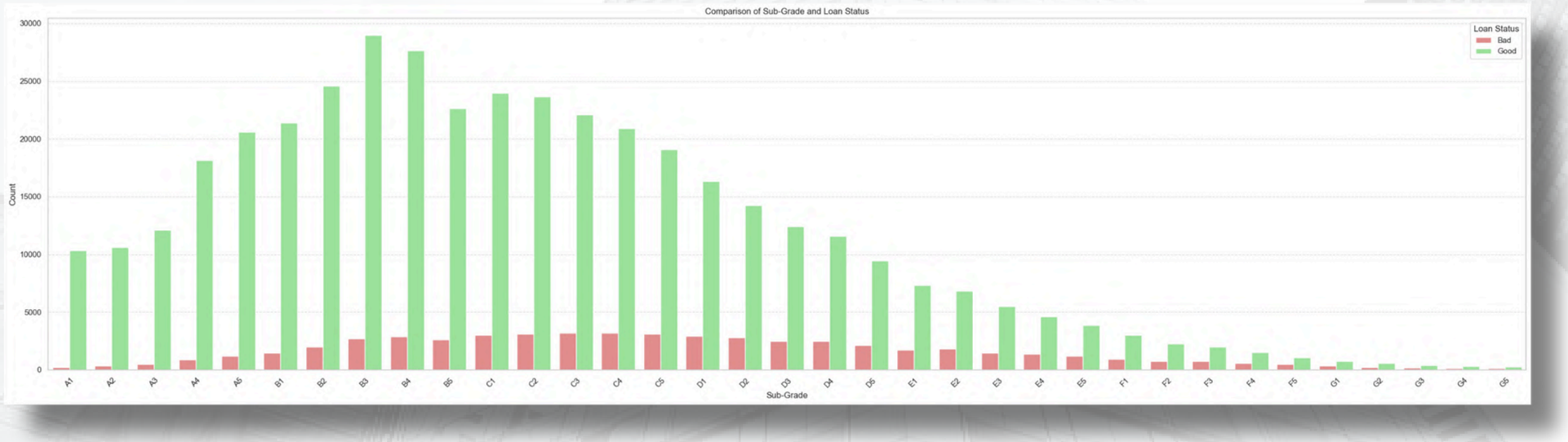
- Popular Grades: Grades B and C dominate in loan counts.
- Loan Quality: Higher grades (A, B, C) have more "Good" loans, while lower grades (E, F, G) show a higher proportion of "Bad" loans.
- Risk Assessment: Lower grades are riskier, reflected in the higher proportion of "Bad" loans.



Loan Status by Sub Grade Analysis

Sub-Grade Analysis:

- Higher Sub-Grades: Most loans are in higher sub-grades (B3 to D3).
- Loan Status: "Good" loans outnumber "Bad" loans across all sub-grades.
- Distribution: B3, C1, and C2 are the most common sub-grades.



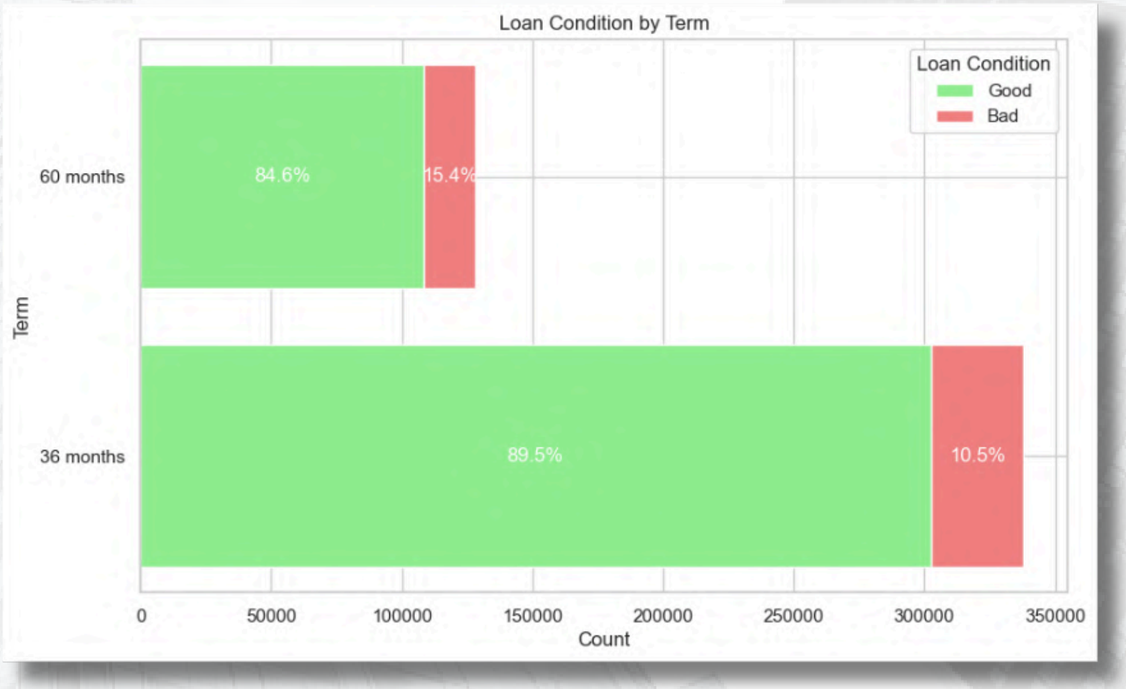
Loan Status by Term Analysis

36 Months Term:

- Loan Quality: 89.5% "Good", 10.5% "Bad".
- Indicates lower risk with shorter-term loans.

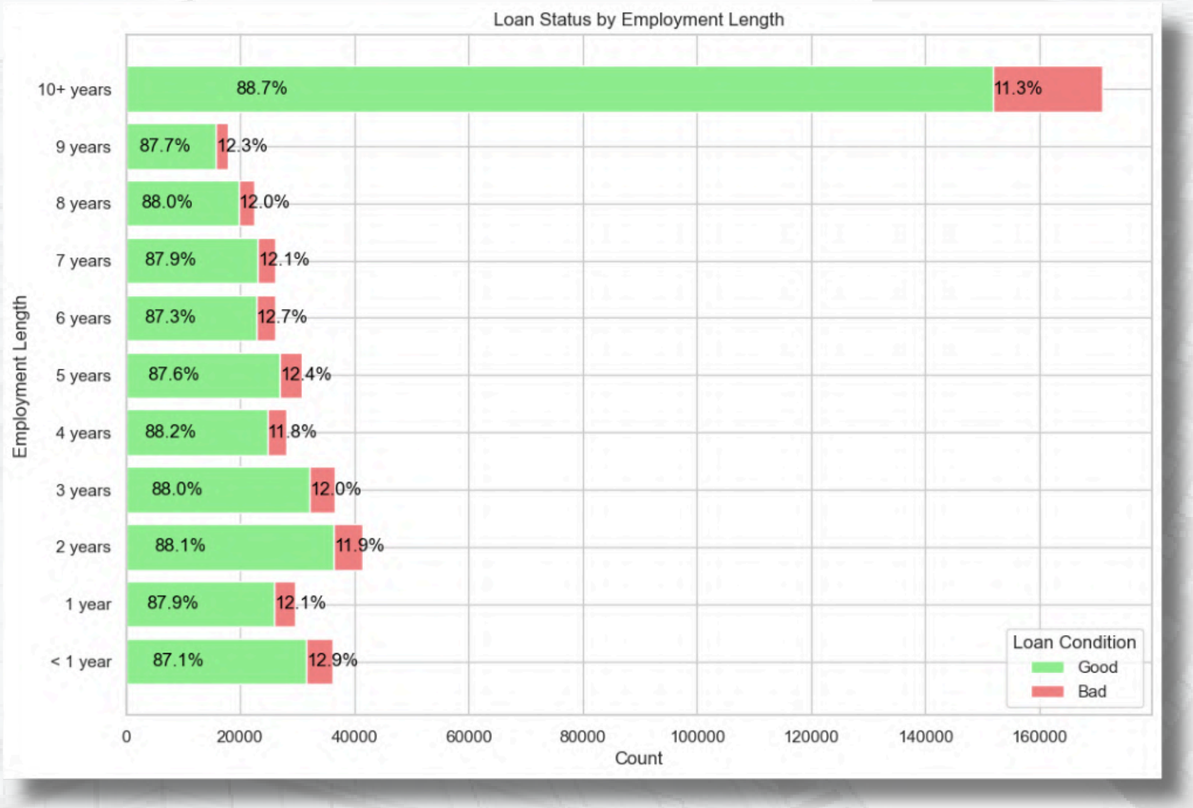
60 Months Term:

- Loan Quality: 84.6% "Good", 15.4% "Bad".
- Higher risk compared to shorter-term loans.



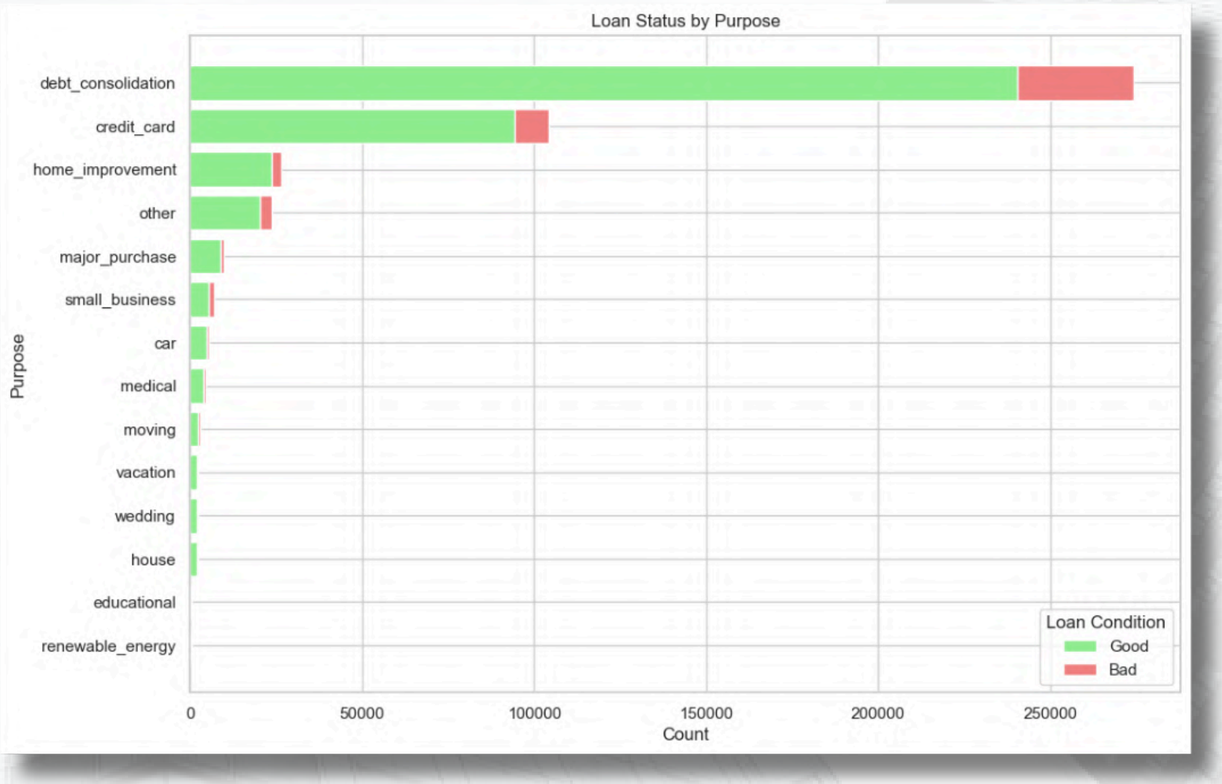
Loan Status by Employment Length Analysis

- **Dominance of Good Loans:** Across all employment lengths, 'Good' loans are the majority, consistently above 87%.
- **Longer Employment Lengths and Loan Status:** The highest proportion of 'Good' loans is found in the '10+ years' employment length category (88.7%).
- **Shorter Employment Lengths:** Even the shortest employment length category ('< 1 year') maintains a high proportion of 'Good' loans at 87.1%, although it has the highest proportion of 'Bad' loans (12.9%) among all categories.
- **Consistent Proportions:** There is a relatively stable distribution of 'Good' and 'Bad' loans across different employment lengths, with only slight variations.



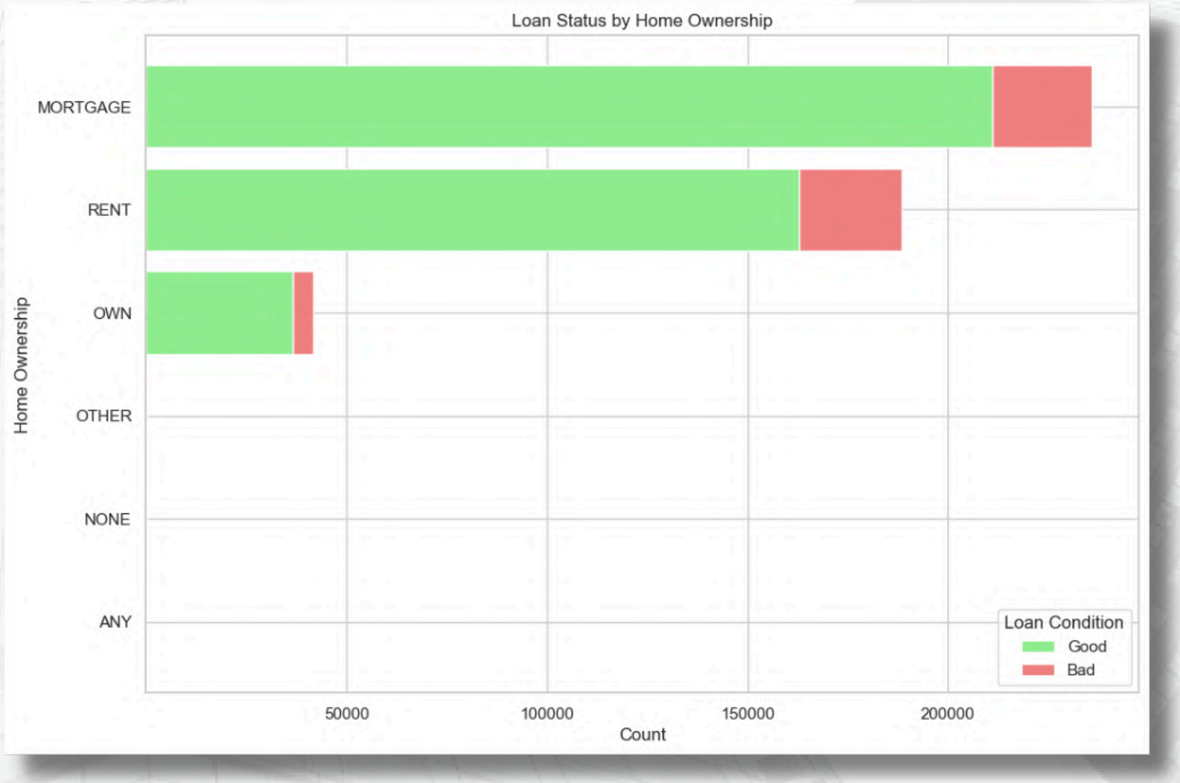
Loan Status by Purpose Analysis

- **Debt Consolidation:** Highest number of loans, mostly 'Good'.
- **Credit Card:** Second largest, dominated by 'Good' loans, relatively higher 'Bad' proportion.
- **Home Improvement and Other:** Significant loan counts, majority 'Good'.
- **Less Popular Purposes:** Few loans, mostly 'Good'.



Loan Status by Home Ownership Analysis

- **Mortgage:** Highest number of loans, majority are 'Good'.
- **Rent:** Second highest, with a significant proportion of 'Bad' loans.
- **Own:** Fewer loans, mostly 'Good'.



Data Preparation

Handling Missing Values

Dropping Columns with > 40% Missing Values:

- Columns like annual_inc_joint, dti_joint, verification_status_joint, etc., have over 40% missing values.
- These columns may lack sufficient data for meaningful analysis or modeling.
- By dropping them, we maintain data integrity and focus on more complete information.

Imputation Approach:

- For columns with < 20% missing values:
  - Numeric columns are imputed using the median.
  - Categorical columns are imputed using the mode.
- This approach preserves data integrity and maximizes available information for analysis, minimizing data loss.

	Missing Values	Missing Percentage (%)
inq_last_12m	466285	100.0000
total_bal_il	466285	100.0000
dti_joint	466285	100.0000
verification_status_joint	466285	100.0000
annual_inc_joint	466285	100.0000
open_acc_6m	466285	100.0000
open_il_6m	466285	100.0000
open_il_12m	466285	100.0000
open_il_24m	466285	100.0000
mths_since_rcnt_il	466285	100.0000
il_util	466285	100.0000
open_rv_24m	466285	100.0000
total_cu_tl	466285	100.0000
inq_fi	466285	100.0000
max_bal_bc	466285	100.0000
all_util	466285	100.0000
open_rv_12m	466285	100.0000
mths_since_last_record	403647	86.5666
mths_since_last_major_derog	367311	78.7739
desc	340304	72.9820
mths_since_last_delinq	250351	53.6906
next_pymnt_d	227214	48.7286
tot_cur_bal	70276	15.0715
tot_coll_amt	70276	15.0715
total_rev_hi_lim	70276	15.0715
emp_title	27588	5.9166
emp_length	21008	4.5054
last_pymnt_d	376	0.0806
revol_util	340	0.0729
collections_12_mths_ex_med	145	0.0311
last_credit_pull_d	42	0.0090
inq_last_6mths	29	0.0062
earliest_cr_line	29	0.0062
delinq_2yrs	29	0.0062
open_acc	29	0.0062
pub_rec	29	0.0062
acc_now_delinq	29	0.0062
total_acc	29	0.0062
title	21	0.0045
annual_inc	4	0.0009

Handling Unique Values

Numerical Columns:

1. **Unnamed: 0, member_id, id:**
   - All unique values, likely serve as unique identifiers without predictive information.
2. **policy_code:**
   - Single unique value, no variation in the dataset, hence not useful for prediction.

	Unique Values	Unique Percentage (%)
Unnamed: 0	466285	100.0000
member_id	466285	100.0000
id	466285	100.0000
total_pymnt	351609	75.4065
total_pymnt_inv	347659	74.5593
total_rec_int	270249	57.9579
tot_cur_bal	220690	47.3294
last_pymnt_amnt	198194	42.5049
total_rec_prncp	172713	37.0402
out_prncp_inv	141189	30.2796
out_prncp	135665	29.0949
revol_bal	58142	12.4692
installment	55622	11.9288
annual_inc	31901	6.8415
recoveries	22773	4.8839
collection_recovery_fee	20275	4.3482
total_rev_hi_lim	14612	3.1337
funded_amnt_inv	9854	2.1133
tot_coll_amt	6321	1.3556
total_rec_late_fee	5808	1.2456
dti	3997	0.8572
funded_amnt	1354	0.2904
loan_amnt	1352	0.2900
revol_util	1269	0.2722
int_rate	506	0.1085
mths_since_last_major_derog	162	0.0347
mths_since_last_delinq	145	0.0311
mths_since_last_record	123	0.0264
total_acc	112	0.0240
open_acc	62	0.0133
inq_last_6mths	28	0.0060
pub_rec	26	0.0056
delinq_2yrs	24	0.0051
collections_12_mths_ex_med	9	0.0019
acc_now_delinq	6	0.0013
policy_code	1	0.0002

Handling Unique Values

Categorical Columns:

1. **url:**
   - All unique values, providing no additional relevant information for prediction.
2. **emp_title, desc:**
   - Many unique values, challenging to extract predictive insights.
3. **title:**
   - Significant unique values, but may lack specificity for loan status prediction.
4. **zip_code, addr_state:**
   - Geographic information, potentially less directly relevant for distinguishing good vs. bad loans.
5. **policy_code:**
   - Single unique value, no variation in the dataset, hence not useful for prediction.
6. **application_type:**
   - Single value, offering limited variation for prediction.

	Unique Values	Unique Percentage (%)
url	466285	100.0000
emp_title	205475	44.0664
desc	124435	26.6865
title	63098	13.5321
zip_code	888	0.1904
earliest_cr_line	664	0.1424
last_credit_pull_d	103	0.0221
next_pymnt_d	100	0.0214
last_pymnt_d	98	0.0210
issue_d	91	0.0195
addr_state	50	0.0107
sub_grade	35	0.0075
purpose	14	0.0030
emp_length	11	0.0024
loan_status	9	0.0019
grade	7	0.0015
home_ownership	6	0.0013
verification_status	3	0.0006
term	2	0.0004
initial_list_status	2	0.0004
pymnt_plan	2	0.0004
application_type	1	0.0002

Feature Engineering - Datetime Data Type

Feature Engineering: Credit Age in Years

- Calculated as the difference between the issue date (issue_d) and earliest credit line date (earliest_cr_line). Represents the length of time the borrower has had credit accounts, providing insights into credit history.

Feature Engineering: Days Since Last Credit Pull

- Computed as the difference between the last credit pull date (last_credit_pull_d) and the issue date (issue_d). Indicates how recently the borrower's credit history was reviewed, reflecting current creditworthiness.

Feature Engineering: Days Since Last Payment

- Derived from the difference between the issue date (issue_d) and the last payment date (last_pymnt_d).Indicates the recency of the borrower's last payment, reflecting current payment behavior.

Reasons for Dropping Datetime Features:

- Issue: Datetime features such as earliest_cr_line, last_credit_pull_d, last_pymnt_d, and issue_d cannot be directly used for modeling without conversion to numerical or categorical formats.

Feature Encoding

- **Label Encoding**

```
# Columns to be label encoded
label_encoded_columns = ['initial_list_status', 'pymnt_plan']

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Apply LabelEncoder to each column
for col in label_encoded_columns:
    data[col] = label_encoder.fit_transform(data[col])
```

Reasoning: Label encoding assigns a unique integer to each category. It is suitable for binary categorical variables or those with a small number of unique values where the order does not matter. It simplifies data representation while maintaining the categorical nature of the data.

- **One-Hot Encoding**

```
# Columns to be one-hot encoded
one_hot_encoded_columns = ['term', 'home_ownership', 'verification_status', 'purpose']

# Extracting columns for one-hot encoding
data_subset = data[one_hot_encoded_columns].copy()

# Initialize OneHotEncoder
encoder = OneHotEncoder(sparse=False)

# Fit-transforming the data
encoded_data = encoder.fit_transform(data_subset)
```

Reasoning: One-hot encoding is suitable when categorical variables have no inherent order and are not ordinal. It creates binary columns for each category, preserving distinct categories without imposing a false sense of order.

Feature Encoding

- **Manual Mapping**

```python
# Manual mapping for emp_length
emp_length_mapping = {
    '10+ years': 10,
    '9 years': 9,
    '8 years': 8,
    '7 years': 7,
    '6 years': 6,
    '5 years': 5,
    '4 years': 4,
    '3 years': 3,
    '2 years': 2,
    '1 year': 1,
    '< 1 year': 0
}

# Apply mapping to emp_length column
data_mapped_emp_length = data.replace({'emp_length': emp_length_mapping})
```

```python
# Manual mapping for grade
grade_mapping = {'A': 1, 'B': 2, 'C': 3, 'D': 4, 'E': 5, 'F': 6, 'G': 7}

# Apply mapping to grade column
data_mapped_grade = data_mapped_emp_length.replace({'grade': grade_mapping})
```

```python
# Manual mapping for sub_grade
sub_grade_mapping = {
    'A1': 1, 'A2': 2, 'A3': 3, 'A4': 4, 'A5': 5,
    'B1': 6, 'B2': 7, 'B3': 8, 'B4': 9, 'B5': 10,
    'C1': 11, 'C2': 12, 'C3': 13, 'C4': 14, 'C5': 15,
    'D1': 16, 'D2': 17, 'D3': 18, 'D4': 19, 'D5': 20,
    'E1': 21, 'E2': 22, 'E3': 23, 'E4': 24, 'E5': 25,
    'F1': 26, 'F2': 27, 'F3': 28, 'F4': 29, 'F5': 30,
    'G1': 31, 'G2': 32, 'G3': 33, 'G4': 34, 'G5': 35,
}

# Apply mapping to grade column
data_mapped_sub_grade = data_mapped_grade.replace({'sub_grade': sub_grade_mapping})
```

Reasoning: Manual mapping is appropriate when categorical variables exhibit a clear order or hierarchy, such as grades (A to G) or employment lengths ("< 1 year" to "10+ years"). This preserves the ordinal relationship in the data.

Handling Outliers

```python
# Define a function to remove outliers using IQR
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]

numerical_columns = [
    'loan_amnt', 'int_rate', 'annual_inc', 'dti', 'delinq_2yrs', 'inq_last_6mths',
    'open_acc', 'pub_rec', 'revol_bal', 'revol_util', 'total_acc', 'out_prncp',
    'total_pymnt', 'total_rec_int', 'total_rec_late_fee', 'recoveries',
    'last_pymnt_amnt', 'collections_12_mths_ex_med', 'acc_now_delinq',
    'tot_coll_amt', 'tot_cur_bal', 'total_rev_hi_lim', 'credit_age_years',
    'days_since_last_credit_pull', 'days_since_last_payment'
]

# Remove outliers for each column
for col in numerical_columns:
    data = remove_outliers(df_encoded, col)
```

Outlier Removal Method:

- We employed the IQR method to identify outliers for numerical columns such as loan_amnt, int_rate, annual_inc, and others. This method calculates the lower and upper bounds based on the quartiles of the data distribution.

Impact of Outlier Removal:

- After removing outliers, the dataset size decreased from 466,285 rows to 460,712 rows. This reduction indicates the removal of data points that were unusually extreme, which could potentially skew our analysis and modeling results.

Reasons for Outlier Handling:

- Outliers can distort statistical analyses and machine learning models by affecting measures of central tendency and dispersion. Removing them helps in producing more accurate and reliable results.

Numerical Data Preprocessing Analysis

```python
# Function to replace non-positive values with a small positive value
def replace_non_positive(x):
    return np.where(x <= 0, 1e-6, x)

# Select numerical features for preprocessing
numerical_features = df_encoded.select_dtypes(include=['int64', 'float64', 'int32']).columns

# Initialize pipeline for numerical features
numerical_pipeline = Pipeline([
    ('replace_non_positive', FunctionTransformer(replace_non_positive, validate=True)),
    ('log_transform', FunctionTransformer(np.log1p, validate=True)),
    ('scaler', StandardScaler())
])

# ColumnTransformer specifically for numeric features
preprocessor = ColumnTransformer([
    ('numeric', numerical_pipeline, numerical_features)
])

# Fit-transform numerical data
X_transformed = preprocessor.fit_transform(df_encoded[numerical_features])
```

Handling Non-Positive Values:

- We replaced non-positive values (<= 0) with a small positive value (1e-6) to ensure numerical stability and prevent issues like zero division during transformations.

Logarithmic Transformation:

- Logarithmic transformation (log1p) was applied to normalize the skewed distribution of numerical features. This transformation helps in reducing the impact of outliers and making the data more Gaussian-like.

Standardization:

- Standard scaling (StandardScaler) was used to standardize the transformed numerical data. This step ensures that all features are on the same scale, preventing features with larger ranges from dominating the model training process.

Data Splitting & Handling Imbalanced

```
# Assuming transformed_df contains your preprocessed and transformed data
X = transformed_df.drop('loan_condition', axis=1)  # Features
y = transformed_df['loan_condition']  # Target

# Splitting the data into train and test sets (70:30 split)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Define RandomUnderSampler
undersampler = RandomUnderSampler(random_state=42)

# Undersampling on Training Data
X_train_resampled, y_train_resampled = undersampler.fit_resample(X_train, y_train)
```

Data Splitting:

- The preprocessed data was split into training and testing sets using a 80:20 ratio. This approach facilitates robust model training and evaluation.

Undersampling Strategy:

- The RandomUnderSampler technique was applied exclusively to the training set. This method randomly reduces instances in the majority class (i.e., "good" loans) to align with the minority class (i.e., "bad" loans). By balancing class distribution, undersampling mitigates bias towards the majority class, potentially enhancing predictive accuracy.

Data Modeling & Evaluation

Rakamin Academy

Model Comparison & Cross-Validation

```python
# Define models
models = {
    'Logistic Regression': LogisticRegression(random_state=42, solver='liblinear'),
    'Random Forest': RandomForestClassifier(random_state=42),
    'Gradient Boosting': GradientBoostingClassifier(random_state=42),
    'XGBoost': XGBClassifier(random_state=42),
    'LightGBM': LGBMClassifier(random_state=42),
    'CatBoost': CatBoostClassifier(random_state=42, verbose=0)
}
```

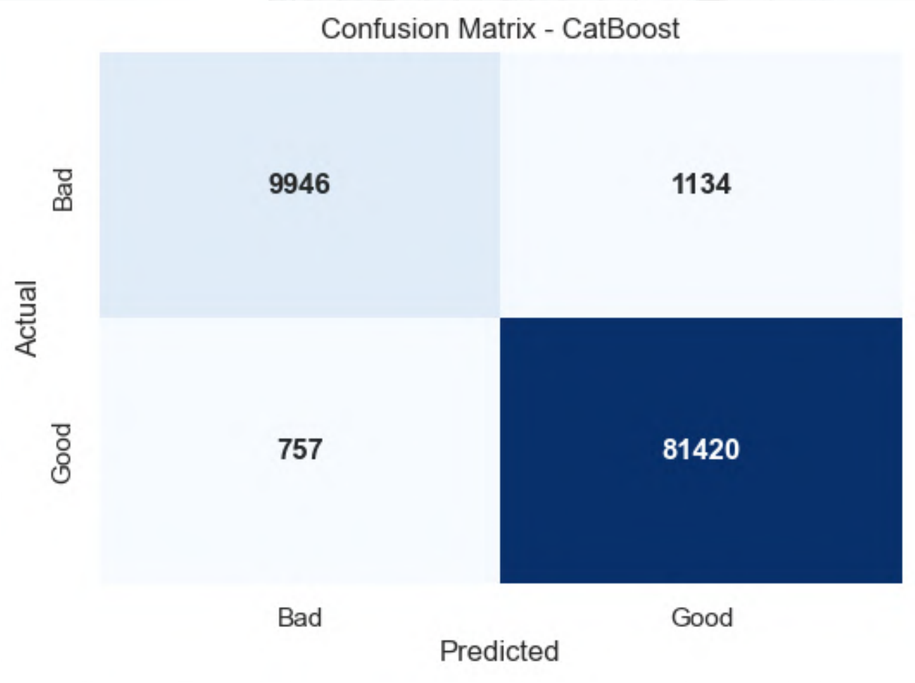
	Model	Mean ROC-AUC	CV Scores
0	Logistic Regression	0.972207	[0.9708095034746623, 0.9725735226062107, 0.9732870555658748, 0.9720979658846681, 0.9722680838839413]
1	Random Forest	0.962798	[0.9617571381062626, 0.9611794256067334, 0.9640336276599344, 0.9646503156143741, 0.9623714769063807]
2	Gradient Boosting	0.958229	[0.9579954003090717, 0.9574322477526559, 0.9591576488851525, 0.9586688088485644, 0.9578901656612084]
3	XGBoost	0.984771	[0.9833366270337484, 0.9844798283190574, 0.9851283426112276, 0.9859020494348686, 0.9850073222892527]
4	LightGBM	0.982170	[0.9811661359085475, 0.9819872120166447, 0.9823086797304016, 0.9832350010564096, 0.9821536467809378]
5	CatBoost	0.987350	[0.9865970649044093, 0.9869178560052905, 0.9876888627605679, 0.9881910244110436, 0.9873543745411598]

Model Comparison on Test Data

Model	Precision (Bad)	Recall (Bad)	F1-Score (Bad)	Precision (Good)	Recall (Good)	F1-Score (Good)	Accuracy	ROC-AUC Score
Logistic Regression	0.82	0.87	0.85	0.98	0.97	0.98	0.96	0.9701
Random Forest	0.81	0.83	0.82	0.98	0.97	0.98	0.96	0.9620
Gradient Boosting	0.86	0.81	0.83	0.97	0.98	0.98	0.96	0.9570
XGBoost	0.89	0.89	0.89	0.99	0.99	0.99	0.97	0.9844
LightGBM	0.92	0.87	0.89	0.98	0.99	0.99	0.98	0.9817
CatBoost	**0.93**	**0.90**	**0.91**	**0.99**	**0.99**	**0.99**	**0.98**	**0.9872**

Catboost Classifier (Before Tuning)

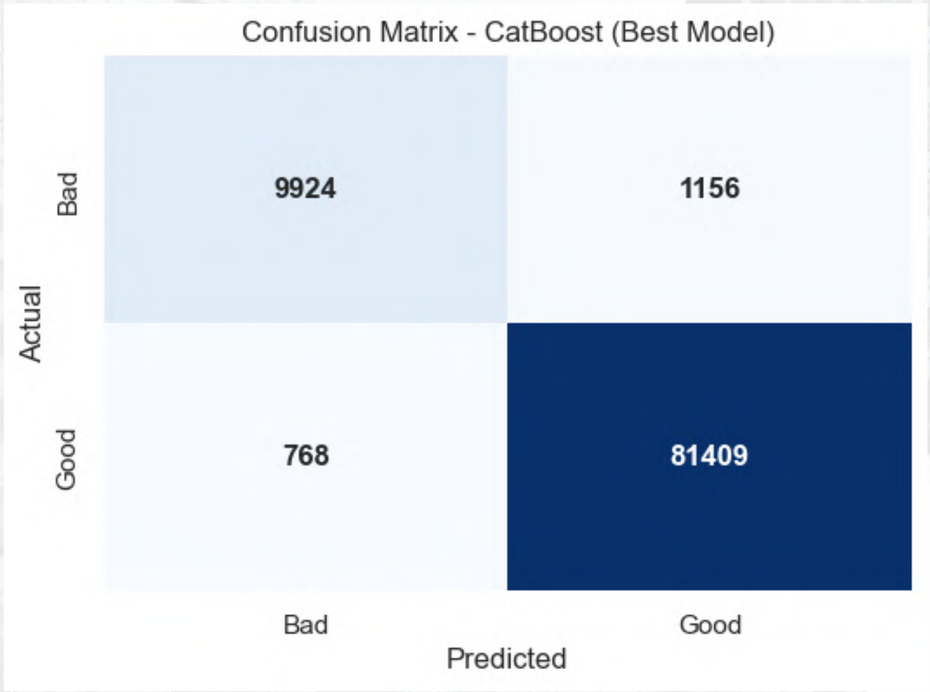
Model	Precision (Bad)	Recall (Bad)	F1-Score (Bad)	Precision (Good)	Recall (Good)	F1-Score (Good)	Accuracy	ROC-AUC Score
CatBoost	**0.93**	**0.90**	**0.91**	**0.99**	**0.99**	**0.99**	**0.98**	**0.9872**



Catboost Classifier (After Tuning)

Model	Precision (Bad)	Recall (Bad)	F1-Score (Bad)	Precision (Good)	Recall (Good)	F1-Score (Good)	Accuracy	ROC-AUC Score
CatBoost	**0.93**	**0.90**	**0.91**	**0.99**	**0.99**	**0.99**	**0.98**	**0.9864**

{'depth': 8,
'iterations': 400,
'l2_leaf_reg': 5,
'learning_rate': 0.1}



Conclusion

Conclusion

Model Selection:

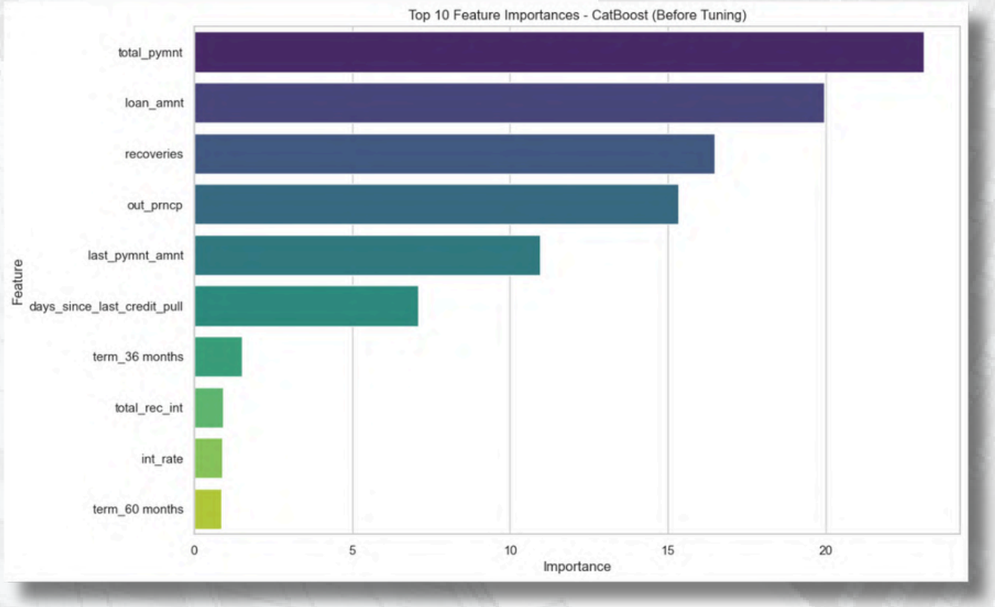
- Six machine learning models were evaluated: Logistic Regression, Random Forest, Gradient Boosting, XGBoost, LightGBM, and CatBoost.
- ROC-AUC was the primary metric for model evaluation.
- CatBoost demonstrated the highest performance with a ROC-AUC Score of 0.9872.

Hyperparameter Tuning:

- Performed on CatBoost to enhance model performance.
- Best cross-validation ROC-AUC Score achieved: 0.9870.
- Test ROC-AUC Score after tuning: 0.9864 (slightly lower than initial model).

Final Model:

- The initial CatBoost model was selected for final implementation due to its superior ROC-AUC performance.



Documentation

Documentation

Folder Google Drive Link:

https://drive.google.com/drive/folders/1KxVcw58O24bbykAVDYhvUWm4N9ykRGos?usp=sharing

Repository Final Task GitHub Link:

https://github.com/ahmfzui/Final-Task_ID-X-Partners_Data-Scientist_Ahmad-Fauzi.git

Video Presentation Link:

https://drive.google.com/file/d/1mWc0_-nOADuvolQPaGKc06-IbsJWQe0Z/view?usp=sharing

Thank You

Rakamin Academy X id/x partners