Machine Learning Project

Credit Risk Analysis & Predictions

Data Science Project Based Internship

Presented by
Oktavian Dwi Putra

Oktavian Dwi Putra

About Me

Result oriented professional with background in digital marketing, especially SEO and a strong desire to transition into the field of Data Science. Possessing a solid foundation in statistics, machine learning, and data analysis. Eager to apply my analytical mindset, problem solving abilities, and passion for data driven insights to drive meaningful outcomes as a Data Scientist.

Experiences

SEO Specialist cmlabs
Apr 2022 Mar 2023

Improve website visibility and performance in search engines for several clients, including:

- Do keyword research for new content weekly.
- Perform onpage optimization for existing content to improve their performance.
- Monitor the ranking of targeted keywords from existing content.

Background Story

As an intern Data Scientist at ID/X Partners, you will be involved in a project from a **lending company**. You will collaborate with various other departments in this project to provide technology solutions for the company. You are asked to **build a model that can predict credit risk** using a dataset provided by the company which consists of data on loans accepted and rejected.

Beside that, you also need to prepare **visual media** to present solutions to clients. Make sure the visual media you create is clear, easy to read and communicative. Working on this end to end solution can be done in the programming language of your choice while still referring to the Data Science framework/methodology.

Goals:

1. Reducing the percentage of bad loans to below 2.5% (average Indonesia non performing loans percentage).
2. Find out the factors that can predict whether a loan is good or bad.

Objectives:

1. Analyze historical data on good and bad loans to discover insights and patterns.
2. Create a machine learning classification model to predict whether a loan is good or bad.

Data Description (1)

Feature	Description	Type
id	A unique LC assigned ID for the loan listing	Numerical
member_id	A unique LC assigned Id for the borrower member	Numerical
loan_amnt	The listed amount of the loan applied by the borrower	Numerical
funded_amnt	The total amount committed to that loan at that point in time	Numerical
funded_amnt_inv	The total amount committed to that loan by the investors at that point in time	Numerical
term	The number of payments on the loan. Values are in months and can be either 36 or 60	Categorical
int_rate	Interest rate on the loan	Numerical
installment	The monthly payment owed by the borrower if the loan originates	Numerical
grade	LC assigned loan grade	Categorical
sub_grade	LC assigned loan subgrade	Categorical
emp_title	The job title from the borrower when applying for the loan	Categorical
emp_length	Employment length in years	Categorical
home_ownership	The home ownership status from the borrower	Categorical
annual_inc	The self-reported annual income provided by the borrower during registration	Numerical
verification_status	Indicates if the income was verified by LC, not verified, or if the income source was verified	Categorical
issue_d	The month which the loan was funded	Categorical
loan_status	Loan payment status	Categorical
pymnt_plan	Indicates if a payment plan has been put in place for the loan	Categorical
url	URL for the LC page with listing data	Categorical
desc	Loan description provided by the borrower	Categorical

Data Description (2)

purpose	A category provided by the borrower for the loan request	Categorical
title	The loan title provided by the borrower	Categorical
zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application	Categorical
addr_state	The state provided by the borrower in the loan application	Categorical
dti	Total monthly debt payments excluding mortgage and the requested LC loan divided by monthly income	Numerical
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years	Numerical
earliest_cr_line	The date the borrower's earliest reported credit line was opened	Categorical
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)	Numerical
mths_since_last_delinq	The number of months since the borrower's last delinquency	Numerical
mths_since_last_record	The number of months since the last public record	Numerical
open_acc	Number of open trades	Numerical
pub_rec	Number of derogatory public records	Numerical
revol_bal	Total credit revolving balance	Numerical
revol_util	Revolving line utilization rate or the amount of credit the borrower is using relative to all available revolving credit	Numerical
total_acc	The total number of credit lines currently in the borrower's credit file	Numerical
initial_list_status	The initial listing status of the loan	Categorical
out_prncp	Remaining outstanding principal for total amount funded	Numerical
out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors	Numerical
total_pymnt	Payments received to date for total amount funded	Numerical
total_pymnt_inv	Payments received to date for portion of total amount funded by investors	Numerical

Data Description (3)

total_rec_prncp	Principal received to date	Numerical
total_rec_int	Interest received to date	Numerical
total_rec_late_fee	Late fees received to date	Numerical
recoveries	The funds that are recovered by a lender after a borrower has failed to meet their repayment obligations	Numerical
collection_recovery_fee	Post charge off collection fee	Numerical
last_pymnt_d	Last month payment was received	Categorical
last_pymnt_amnt	Last total payment amount received	Numerical
next_pymnt_d	Next scheduled payment date	Categorical
last_credit_pull_d	The most recent month LC pulled credit for this loan	Categorical
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections	Numerical
mths_since_last_major_derog	Months since most recent 90-day or worse rating	Numerical
policy_code	publicly available policy_code=1; new products not publicly available policy_code=2	Numerical
application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers	Categorical
annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration	Numerical
dti_joint	dti for the co-borrowers	Numerical
verification_status_joint	Indicates if the co-borrowers joint income was verified by LC, not verified, or if the income source was verified	Categorical
acc_now_delinq	The number of accounts on which the borrower is now delinquent	Numerical
tot_coll_amt	Total collection amounts ever owed	Numerical
tot_cur_bal	Total current balance of all accounts	Numerical
open_acc_6m	Number of open trades in last 6 months	Numerical

Data Description (4)

open_il_6m	Number of currently active installment trades	Numerical
open_il_12m	Number of installment accounts opened in past 12 months	Numerical
open_il_24m	Number of installment accounts opened in past 24 months	Numerical
mths_since_rcnt_il	Months since most recent installment accounts opened	Numerical
total_bal_il	Total current balance of all installment accounts	Numerical
il_util	Ratio of total current balance to high credit/credit limit on all install acct	Numerical
open_rv_12m	Number of revolving trades opened in past 12 months	Numerical
open_rv_24m	Number of revolving trades opened in past 24 months	Numerical
max_bal_bc	Maximum current balance owed on all revolving accounts	Numerical
all_util	Balance to credit limit on all trades	Numerical
total_rev_hi_lim	Total revolving high credit/credit limit	Numerical
inq_fi	Number of personal finance inquiries	Numerical
total_cu_tl	Number of finance trades	Numerical
inq_last_12m	Number of credit inquiries in past 12 months	Numerical

1. Exploratory Data Analysis (EDA)

1.1. Dataset Info

- Dataset consists of 466285 rows, 74 features and 1 Unnamed: 0 column which is the index.
- Dataset consists of 3 data types: int64, float64, and object.
- The dataset does not have a target variable therefore we need to create it first.
- issue_d, last_pymnt_d, next_pymnt_d, last_credit_pull_d, and earliest_cr_line features should converted into datetime data type.
- There are forty columns that have null values.

```
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                         125983 non-null  object
 21  purpose                      466285 non-null  object
 22  title                        466265 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)
```

1.2. Target Variable

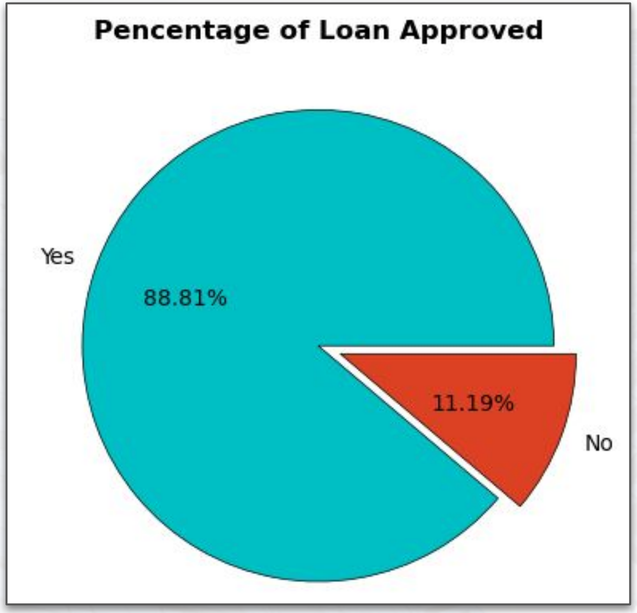
The target variable will be created from the **loan_status** feature, under the conditions:

Good Loan Status:

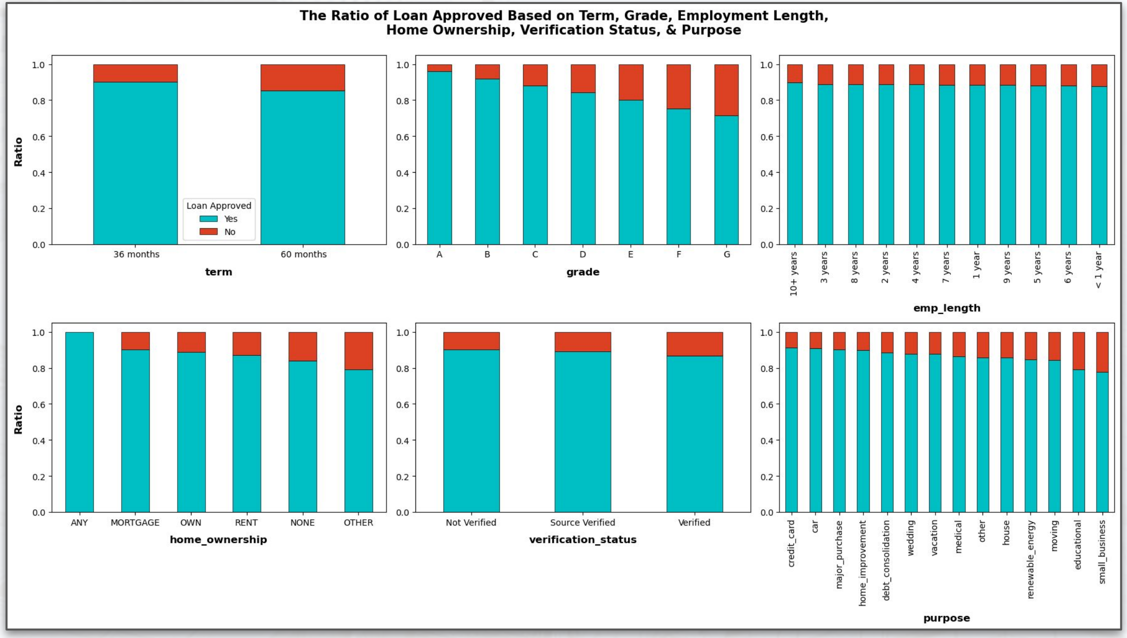
- Fully Paid
- Current
- In Grace Period
- Does not meet the credit policy. Status:Fully Paid

Bad Loan Status:

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



1.3. Univariate Analysis

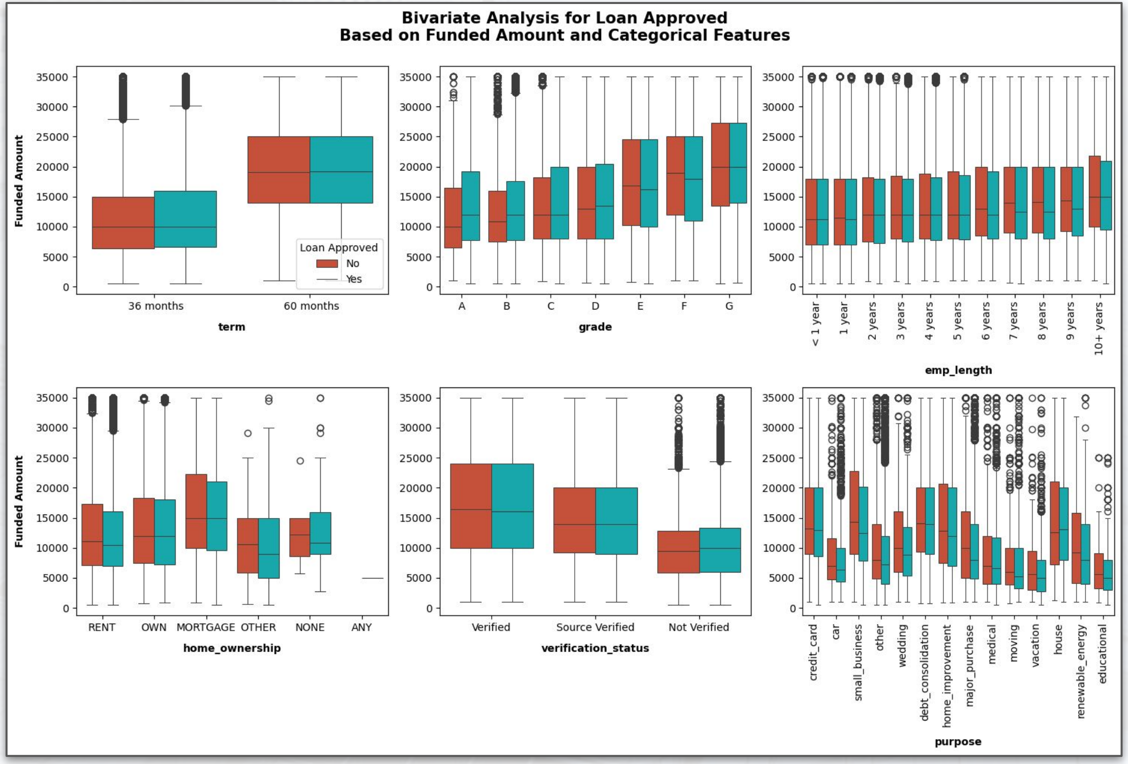


Observation:

- The longer the term, the higher the probability of bad credit.
- Grade A has the lowest probability of bad credit and Grade G has the highest probability.
- Each emp_length has a fairly similar bad credit ratio with the lowest being 10+ years and the highest being < 1 year.
- MORTGAGE home_ownership has a lower probability of bad credit than OWN and RENT.
- Income with Verified status actually has the highest bad credit ratio.
- The lowest probability of bad credit is when the loan is used for a credit card and the highest is for small businesses.

1.4. Univariate Analysis

Rakamin Academy



Observation:

- The longer the term, the higher funded amount.
- Grade B has the lowest funded amount and Grade G has the highest.
- The longer emp_length, the higher funded amount.
- The highest funded amount is when home_ownership is MORTGAGE instead of OWN or RENT.
- Income with Verified status has the highest funded amount and Not Verified status has the lowest.
- The highest funded amount is when the loan is used for a small business and the lowest is for vacation.

2. Data Preparation

2.1. Handle Missing Values

Rakamin Academy

	Features	Null Values
0	emp_length	21008
1	annual_inc	4
2	delinq_2yrs	29
3	earliest_cr_line	29
4	inq_last_6mths	29
5	mths_since_last_delinq	250351
6	mths_since_last_record	403647
7	open_acc	29
8	pub_rec	29
9	revol_util	340
10	total_acc	29
11	last_pymnt_d	376
12	next_pymnt_d	227214
13	last_credit_pull_d	42
14	collections_12_mths_ex_med	145
15	mths_since_last_major_derog	367311
16	acc_now_delinq	29
17	tot_coll_amt	70276
18	tot_cur_bal	70276
19	total_rev_hi_lim	70276

Observation:

There are several treatment that will be done to handle missing values such as:

- Impute the null values with < 1 year for the emp_length column because we assumed that they don't have any employment experience.
- Impute the null values with mode for the earliest_cr_line, last_pymnt_d, and last_credit_pull_d columns.
- Impute the null values with median for the annual_inc, delinq_2yrs, inq_last_6mths, open_acc, pub_rec, total_acc, collections_12_mths_ex_med, and acc_now_delinq columns because they have right-skewed distributions.
- Impute the null values with mean for the revol_util column because it has almost symmetric distribution.
- Remove the mths_since_last_delinq, mths_since_last_record, next_pymnt_d, mths_since_last_major_derog, tot_coll_amt, tot_cur_bal, and total_rev_hi_lim columns because they have too many missing values.

2.2. Handle Duplicate Data

```
df.duplicated().sum()

0
```

- Dataset does not have duplicated data.

2.3. Feature Engineering

There are 3 new features that are made from date related features, namely:

- **loan_duration:** Calculate the duration of the loan by subtracting the issue_date from the last_pymnt_date. This can give an indication of how long the borrower took to repay the loan.
- **credit_hist_len:** Calculate the length of the borrower's credit history by subtracting earliest_cr_line from the issue_date. This can provide insights into the borrower's creditworthiness based on the length of their credit history.
- **credit_report_age:** Calculate the age of the credit report by subtracting last_credit_pull_date from the current date or a reference date. This can indicate how recently the credit report was updated.

2.4. Feature Encoding

2.4.1. Label Encoding

```
# Import library
from sklearn.preprocessing import LabelEncoder

# Perform label encoding
data['term'] = LabelEncoder().fit_transform(data['term'])
data['grade'] = LabelEncoder().fit_transform(data['grade'])
data['sub_grade'] = LabelEncoder().fit_transform(data['sub_grade'])
data['emp_length'] = data['emp_length'].map({'< 1 year': 0, '1 year': 1, '2 years': 2, '3 years': 3,
                                             '4 years': 4, '5 years': 5, '6 years': 6, '7 years': 7,
                                             '8 years': 8, '9 years': 9, '10+ years': 10})
data['pymnt_plan'] = LabelEncoder().fit_transform(data['pymnt_plan'])
data['initial_list_status'] = LabelEncoder().fit_transform(data['initial_list_status'])
```

Observation:

The features that will be encoded with label encoding method are the features that only have 2 unique values or ordinal data.

2.4.2. One-Hot Encoding

```
# Perform one-hot encoding
for cat in ['home_ownership', 'verification_status', 'purpose', 'addr_state']:
  df1 = pd.get_dummies(data[cat], prefix=cat)
  data  = data.drop(cat, axis = 1)
  data  = data.join(df1)
```

The features that will be encoded with one-hot encoding method are the features that have nominal data.

2.5. Feature Selection

2.5.1. Mutual Information

	Features	MI Scores
0	recoveries	1.276742e-01
1	total_rec_prncp	1.257113e-01
2	collection_recovery_fee	1.204945e-01
3	purpose_debt_consolidation	7.059912e-02
4	home_ownership_MORTGAGE	6.415762e-02
5	last_pymnt_amnt	6.185230e-02
6	total_pymnt	5.045095e-02
7	loan_duration	4.939740e-02
8	total_pymnt_inv	4.853301e-02
9	home_ownership_RENT	4.018626e-02
10	out_prncp	3.510828e-02
11	out_prncp_inv	3.447049e-02
12	initial_list_status	3.270390e-02
13	verification_status_Verified	3.221552e-02

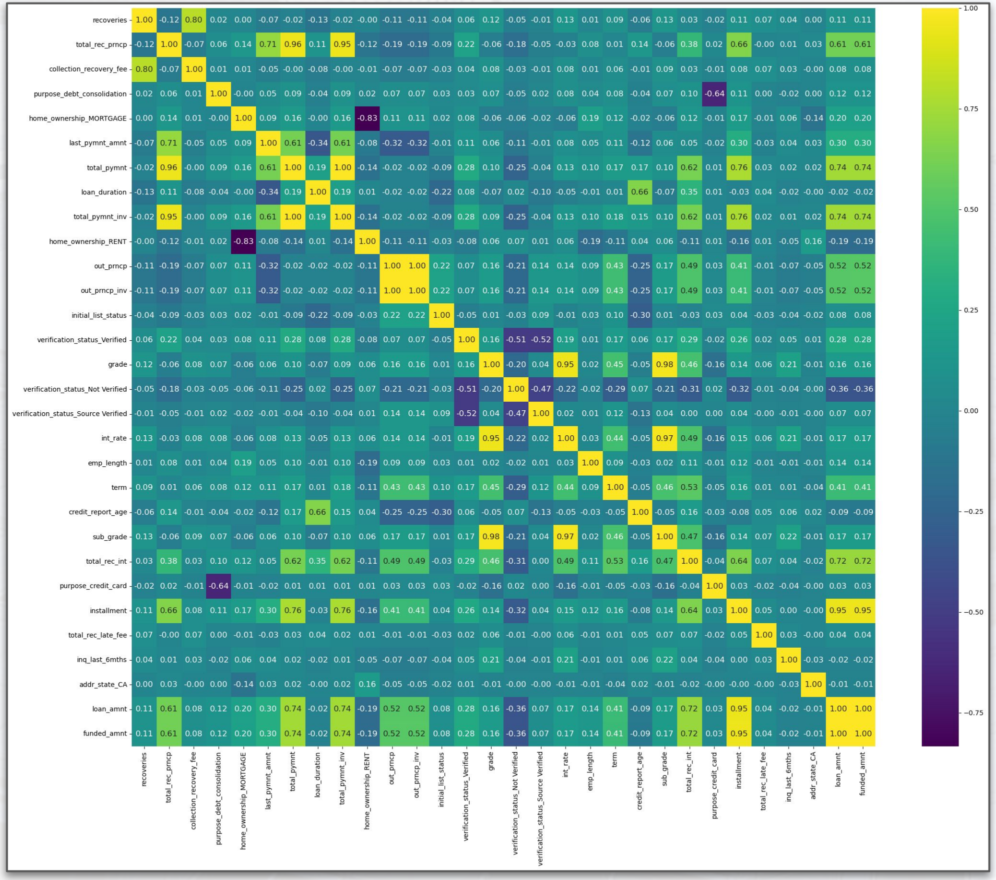
14	grade	0.030208
15	verification_status_Not Verified	0.027228
16	verification_status_Source Verified	0.026101
17	int_rate	0.020517
18	emp_length	0.020411
19	term	0.019971
20	credit_report_age	0.019022
21	sub_grade	0.017057
22	total_rec_int	0.014802
23	purpose_credit_card	0.014538
24	installment	0.012251
25	total_rec_late_fee	0.010199
26	inq_last_6mths	0.009543
27	addr_state_CA	0.006007
28	loan_amnt	0.005659
29	funded_amnt	0.005485

Observation:

For feature selection, we will first calculate the mutual information score for each feature and select the top 30 features that contain useful information for predicting the target variable.

After that, we will calculate the Pearson correlation to see whether among the 30 features there is multicollinearity or high correlation (> 0.7) or not and choose the top 20 features among them.

2.5.2. Pearson Correlation



Observation:

From the heatmap, we can see there are features that have multicollinearity or high correlation (> 0.7) between each other. Therefore based on mutual information score, we will choose the recoveries, total_rec_prncp, out_prncp, grade, and total_rec_int features among the features that have multicollinearity.

We will also drop the addr_state_CA feature, because we only need 20 features for modeling process.

2.6. Split Data

```
# Divide dataset to feature and target
X = data_final
y = data['loan_approved']

# Perform data split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)
```

Observation:

We will split data to train and test data with the 70:30 proportion and random state = 42.

2.7. Standardization

```
# Import library
from sklearn.preprocessing import StandardScaler

# Initiate a Standard scaler
scaler = StandardScaler()

# Create list of column to standardize
column_list = ['recoveries', 'total_rec_prncp', 'loan_duration', 'out_prncp', 'grade', 'emp_length',
               'credit_report_age', 'total_rec_int', 'installment', 'inq_last_6mths', 'total_rec_late_fee']

# Perform scaling process
for col in column_list:
    scaler.fit(X_train[[col]])
    X_train[col] = scaler.transform(X_train[[col]])
    X_test[col] = scaler.transform(X_test[[col]])
```

There are 11 features that still need to be standardized so that the scale is uniform.

3. Modeling

3.1. Initiate Algorithms

Rakamin Academy

```
# Import library
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, ExtraTreesClassifier, GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier

# Instantiation machine learning algorithm
lr = LogisticRegression(random_state = 42)
dt = DecisionTreeClassifier(random_state = 42)
rf = RandomForestClassifier(random_state = 42)
ada = AdaBoostClassifier(random_state = 42)
gb = GradientBoostingClassifier(random_state = 42)
et = ExtraTreesClassifier (random_state = 42)

# Create the models list
models = [lr, dt, rf, ada, gb, et]
```

Observation:

There are 6 algorithms that we will use for modeling process that are the Logistic Regression, Decision Tree, Random Forest, Ada Boost, Gradient Boosting, and Extra Trees algorithms.

3.2. Model Training & Validation

We will choose precision as our main metric because we want to minimize the false positive, namely people who were predicted to be able to repay the loans but apparently cannot. This is because the losses from giving loans to people who are unable to repay the loans are much greater than not giving loans to people who are able to pay the loans.

	Model	Acc (Train)	Acc (Test)	Prec (Train)	Prec (Test)	Recall (Train)	Recall (Test)	ROC AUC (Train)	ROC AUC (Test)	Time Elapsed
0	Decision Tree	0.99999	0.98517	1.00000	0.99139	0.99999	0.99190	0.99999	0.96201	6.576643
1	Random Forest	0.99999	0.98384	0.99999	0.98272	1.00000	0.99936	0.99995	0.93044	133.361791
2	GradientBoost	0.97977	0.97889	0.97808	0.97718	0.99963	0.99955	0.91066	0.90781	124.221854
3	ExtraTress	0.99999	0.97783	1.00000	0.97627	0.99999	0.99931	0.99999	0.90394	82.568233
4	Logistic Regression	0.97430	0.97360	0.97372	0.97315	0.99800	0.99778	0.89184	0.89041	2.956187
5	AdaBoost	0.97089	0.97084	0.96915	0.96906	0.99904	0.99903	0.87297	0.87383	37.106158

From the results above, it can be seen that Decision Tree is the best model because it has the highest Prec (Test) and the worst is the Ada Boost model because it has the lowest Prec (Test) compared to other models.

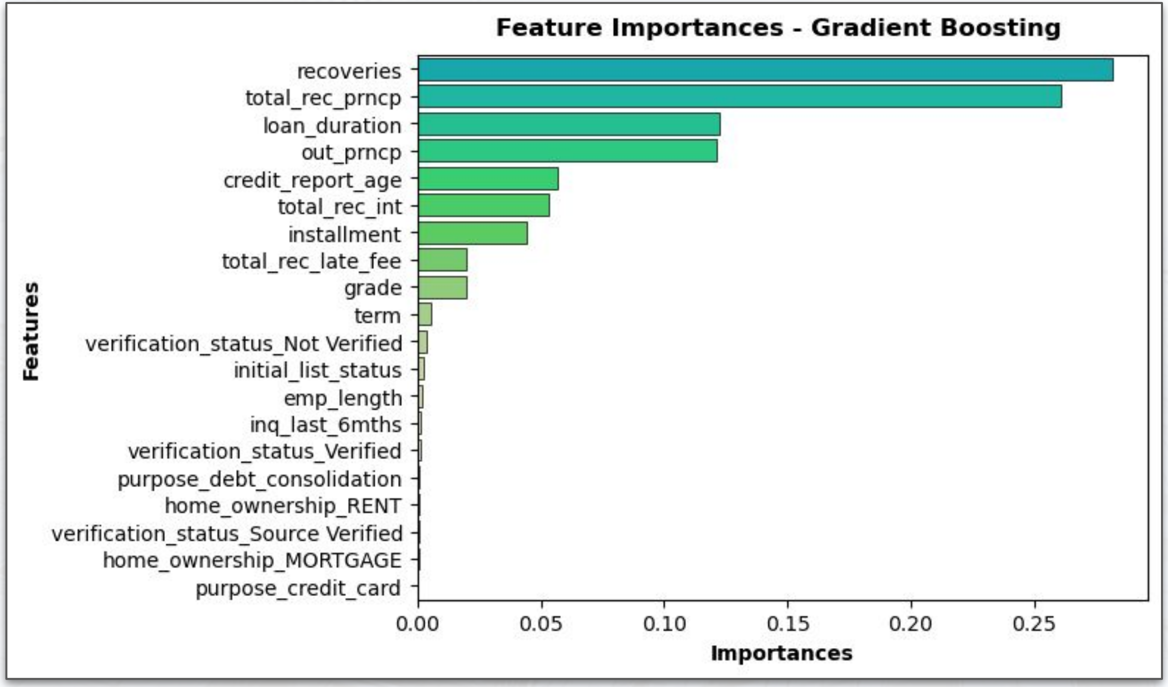
3.3. Hyperparameter Tuning

Rakamin Academy

	Model	Acc (Train)	Acc (Test)	Prec (Train)	Prec (Test)	Recall (Train)	Recall (Test)	ROC AUC (Train)	ROC AUC (Test)	Time Elapsed
0	GradientBoost	0.99226	0.98901	0.99150	0.98831	0.99986	0.99944	0.96583	0.95312	1764.521223
1	Random Forest	0.99988	0.98419	0.99987	0.98301	1.00000	0.99946	0.99948	0.93163	644.425480
2	ExtraTress	0.99604	0.97599	0.99557	0.97413	1.00000	0.99949	0.98228	0.89514	277.672801
3	AdaBoost	0.97570	0.97542	0.97401	0.97365	0.99932	0.99934	0.89354	0.89310	1691.129956
4	Logistic Regression	0.97375	0.97306	0.97308	0.97252	0.99807	0.99783	0.88916	0.88781	64.776840
5	Decision Tree	0.97790	0.95494	0.98354	0.96942	0.99173	0.98014	0.92982	0.86823	11.585753

After hyperparameter tuning there are slightly changes on model performances, it can be seen that Gradient Boosting now is the best model because it has the highest Prec (Test) and the Decision Tree model actually become the model with the worst performance because it has the lowest Prec (Test) compared to other models.

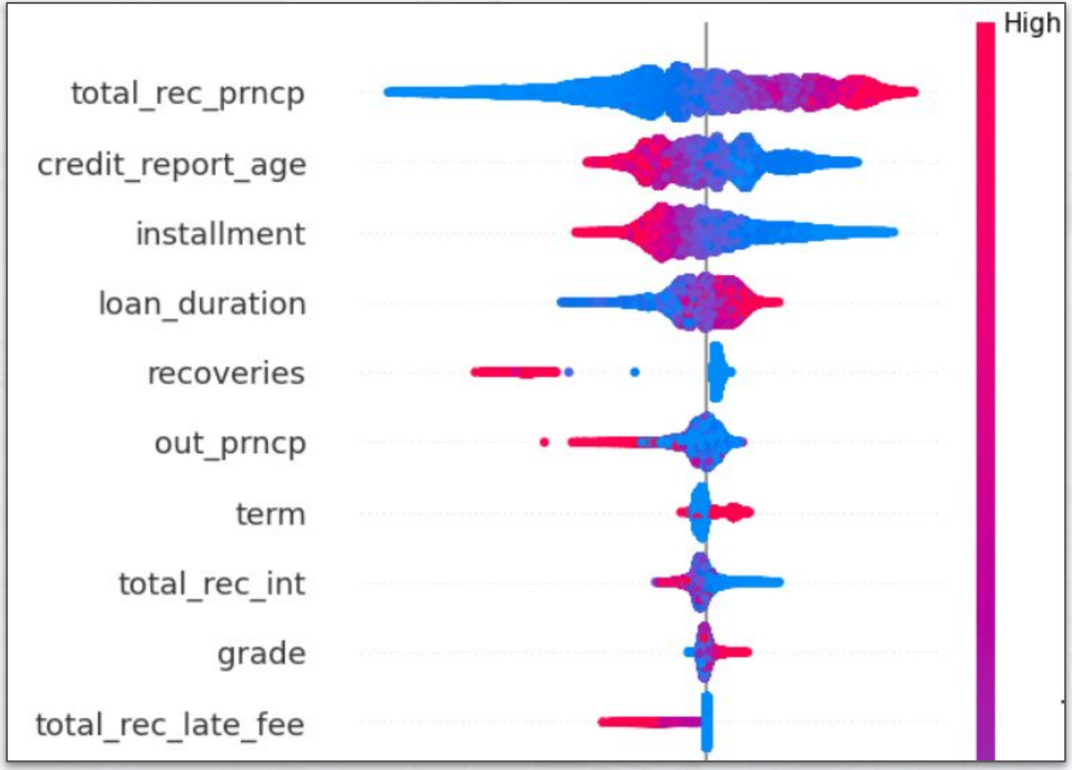
3.4. Feature Importances



Observation:

Based on feature importances from Gradient Boosting model, the top 10 features that have the highest contributions in making accurate predictions are the recoveries, total_rec_prncp, loan_duration, out_prncp, credit_report_age, total_rec_int, installment, total_rec_late_fee, grade, and term features.

3.5. SHAP Values

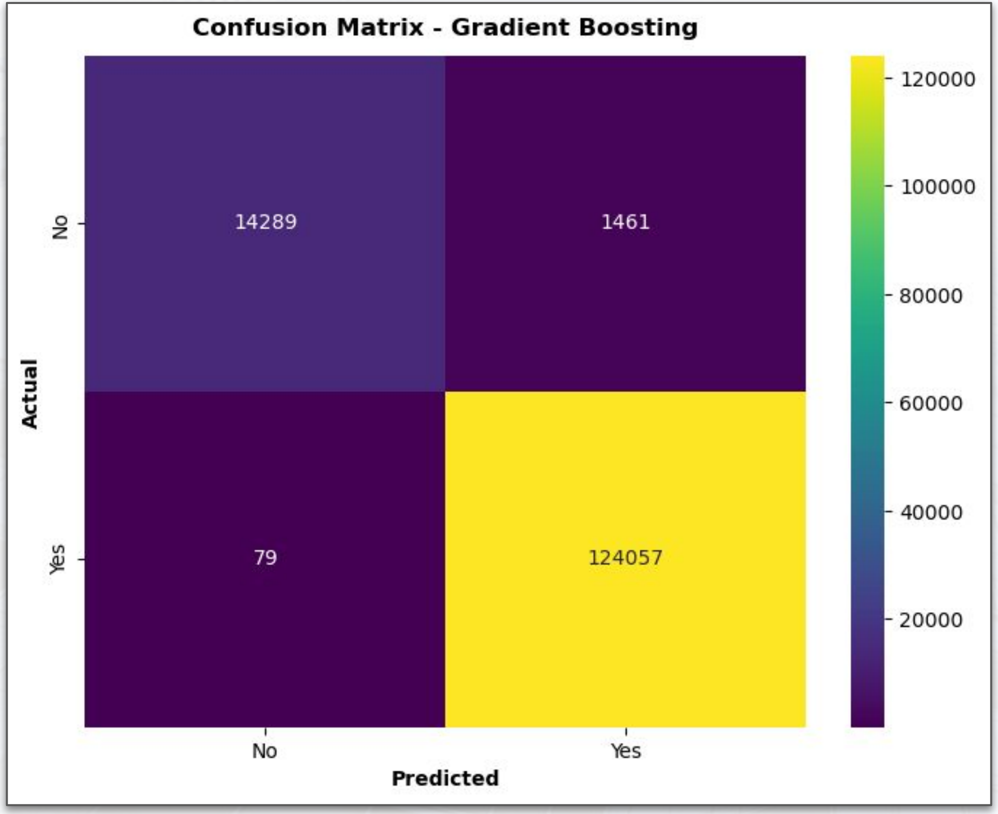


Observation:

From the SHAP values we can see the impact of each feature on the model output. The features that have the higher value tend to be good credit namely total_rec_prncp, loan_duration, term, and grade.

Meanwhile, the features that have the higher value tend to be bad credit namely credit_report_age, installment, recoveries, out_prncp, total_rec_int, and total_rec_late_fee.

3.6. Confusion Matrix



- **True Positive**: Predicted the loan was approved and it turned out to be correct 124,057 times.
- **True Negative**: Predicted the loan was not approved and it turned out to be correct 14,289 times.
- **False Positive**: Predicted the loan was approved and turned out to be wrong by 1,461 times.
- **False Negative**: Predicted the loan was not approved and turned out to be wrong 79 times.

4. Business Simulation

4. Business Simulation

Before Using Machine Learning Model:

Goo Loans Ratio: 0.888

Good Loans:
0.888 * 466,285 = 414,061

Bad Loans:
0.112 * 466,285 = 52,224

After Using Machine Learning Model:

Goo Loans Ratio: 0.988

Good Loans:
0.988 * 466,285 = 414,061

Bad Loans:
0.012 * 466,285 = 52,224

Change Percentage:

Good Loans:
((460,690 - 414,061) / 414,061) * 100% = +11.26%

Bad Loans:
((5,595 - 52,224) / 52,224) * 100% = -89.29%

Conclusion:

After using machine learning, the number of **good loans increased by 11.26%** to 98.8% or the number of **bad loans decreased by 89.29%** to 1.2%.

5. Documentation

5. Documentation

Rakamin Academy

Github

Drive

Linkedin

Thank You

Rakamin Academy X id/x partners