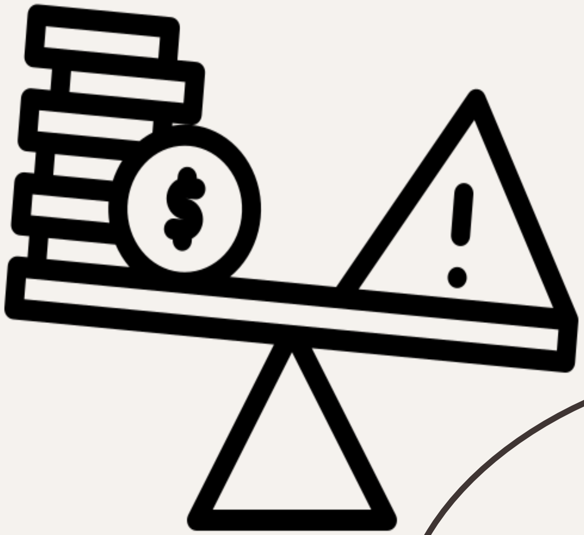
Project-Based Internship Id/x Partners X Rakamin Academy

Muhammad Ifzal Asril

Table of contents

01
Introduction

Credit Risk Prediction



Tools

The tools that will be used are:

- Python
- Pandas
- Matplotlib
- Seaborn
- Google Colab

Dataset

```
data.shape
```

```
(466285, 74)
```

This dataset has 466.285 rows and 74 columns.
With several data types including

1. Float (46 columns)
2. Integer (6 columns)
3. Strings (22 columns)

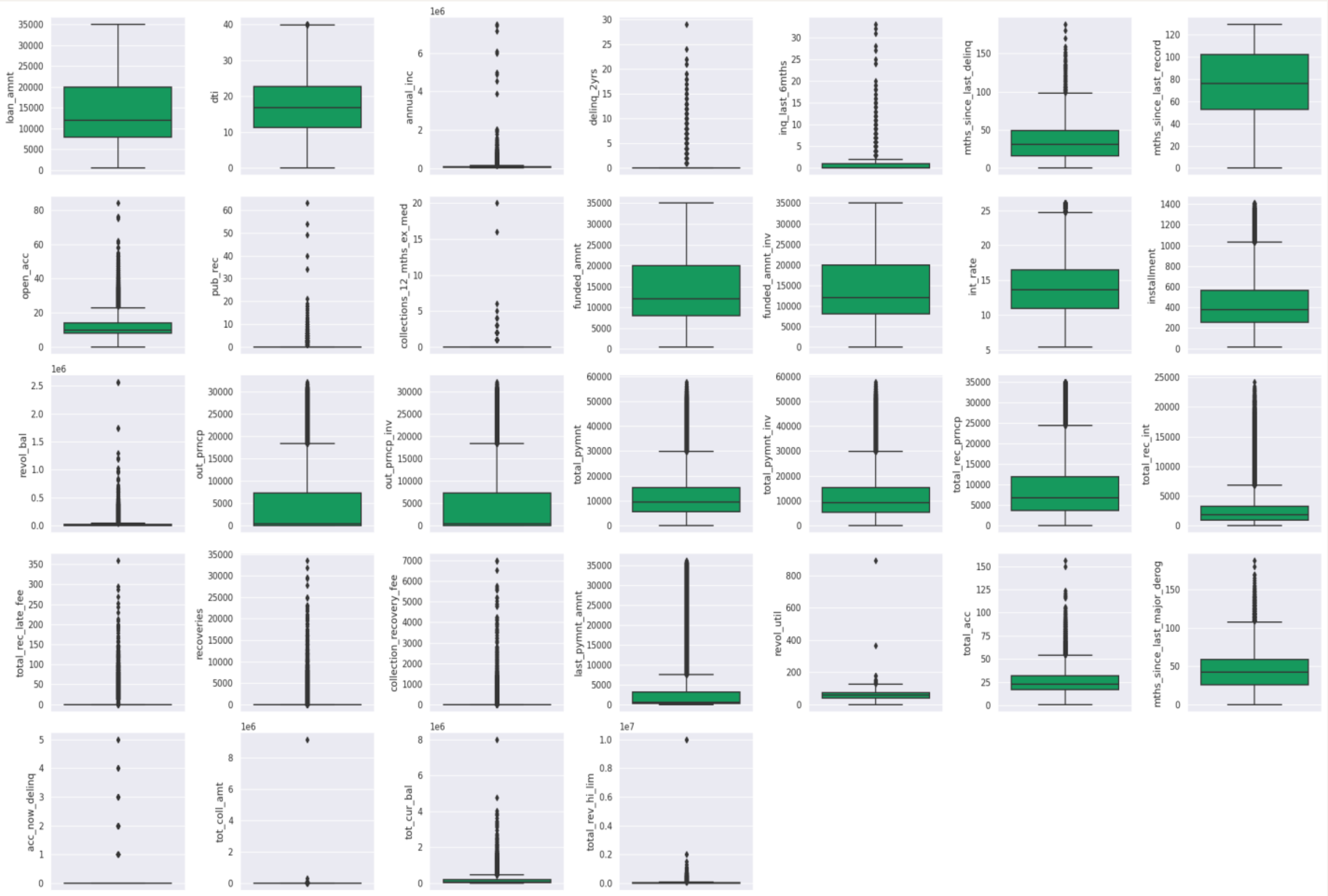
And the target variable is the 'loan_status' column.

02

Data Preprocessing

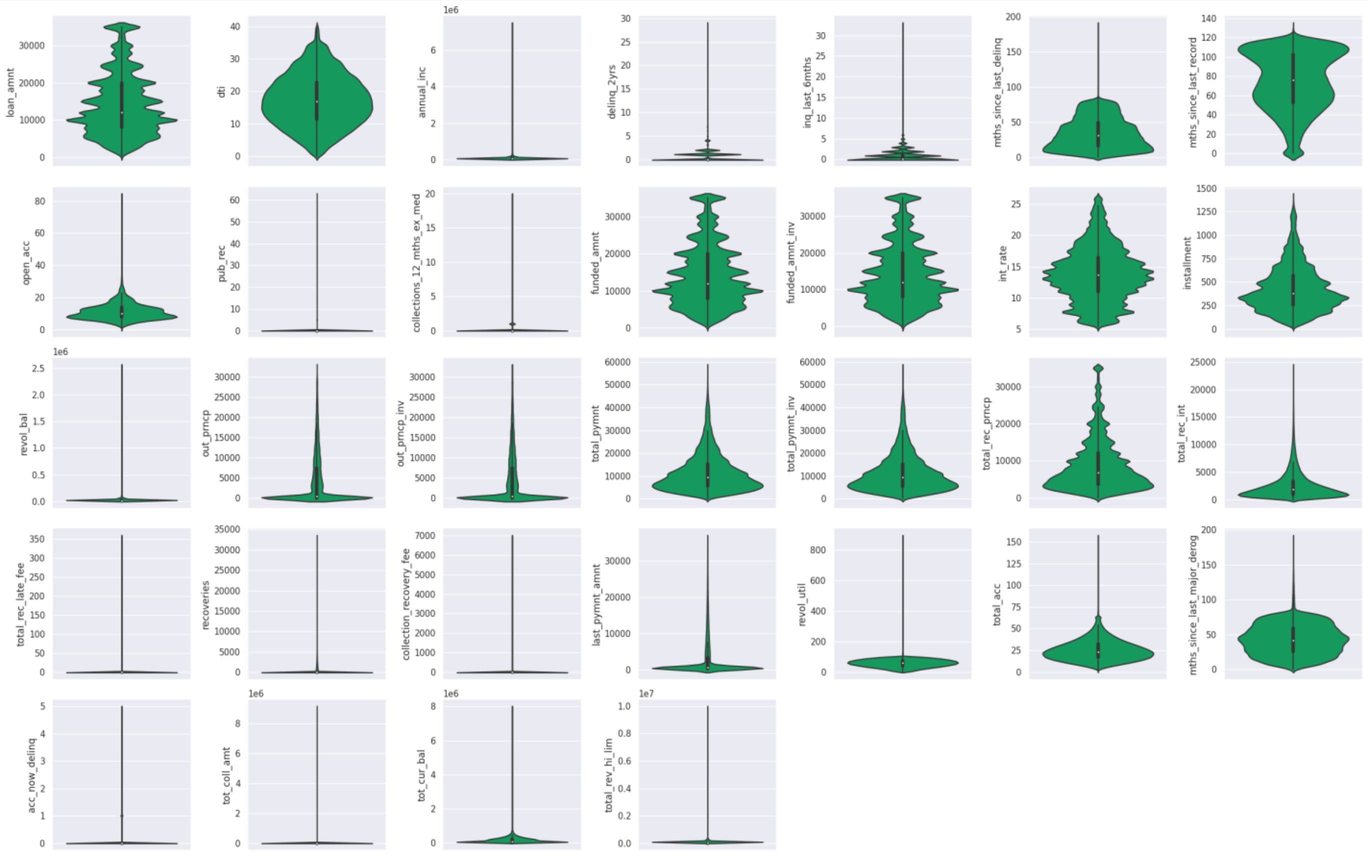
Uni-variate Analysis

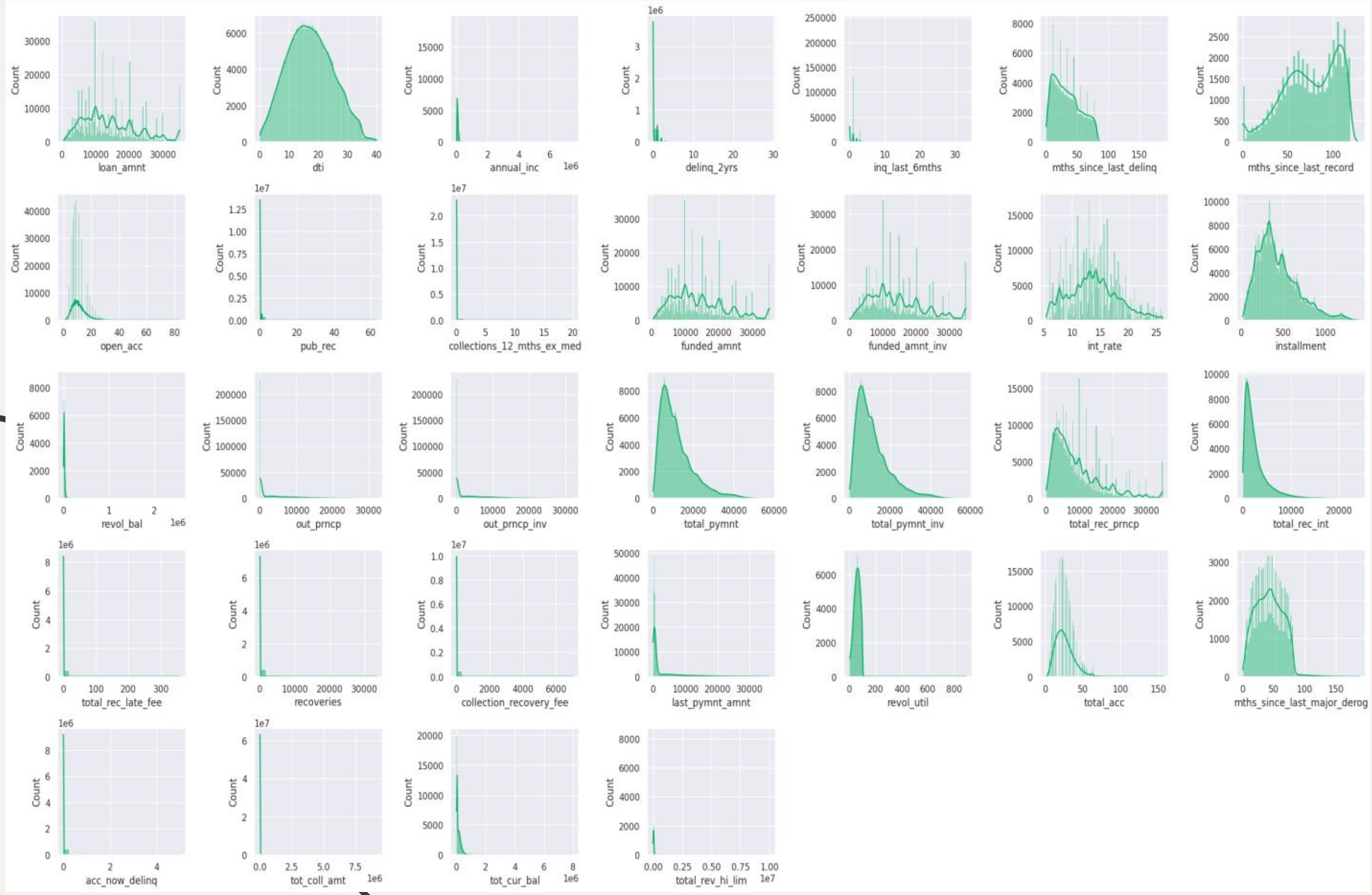
Individual Boxplot



Uni-variate Analysis

Individual Violinplot





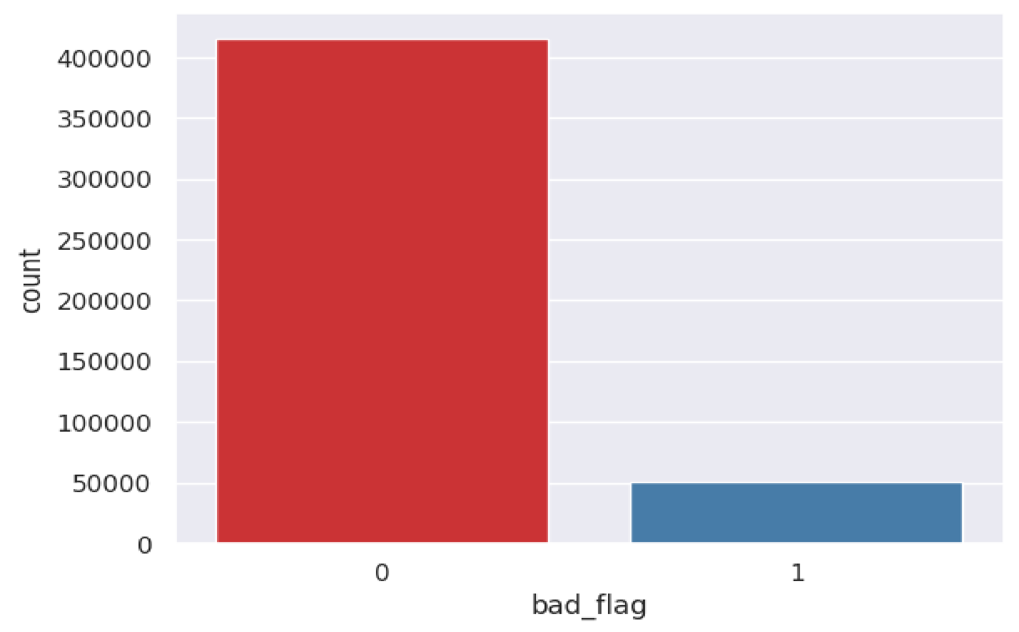
Uni-variate Analysis

Histogram

Variable Target

loan_status
Current = current payment
Charged Off = payment is bad so it is written off
Late = late payment is made
In Grace Period = within the grace period
Fully Paid = payment in full
Default = bad payment

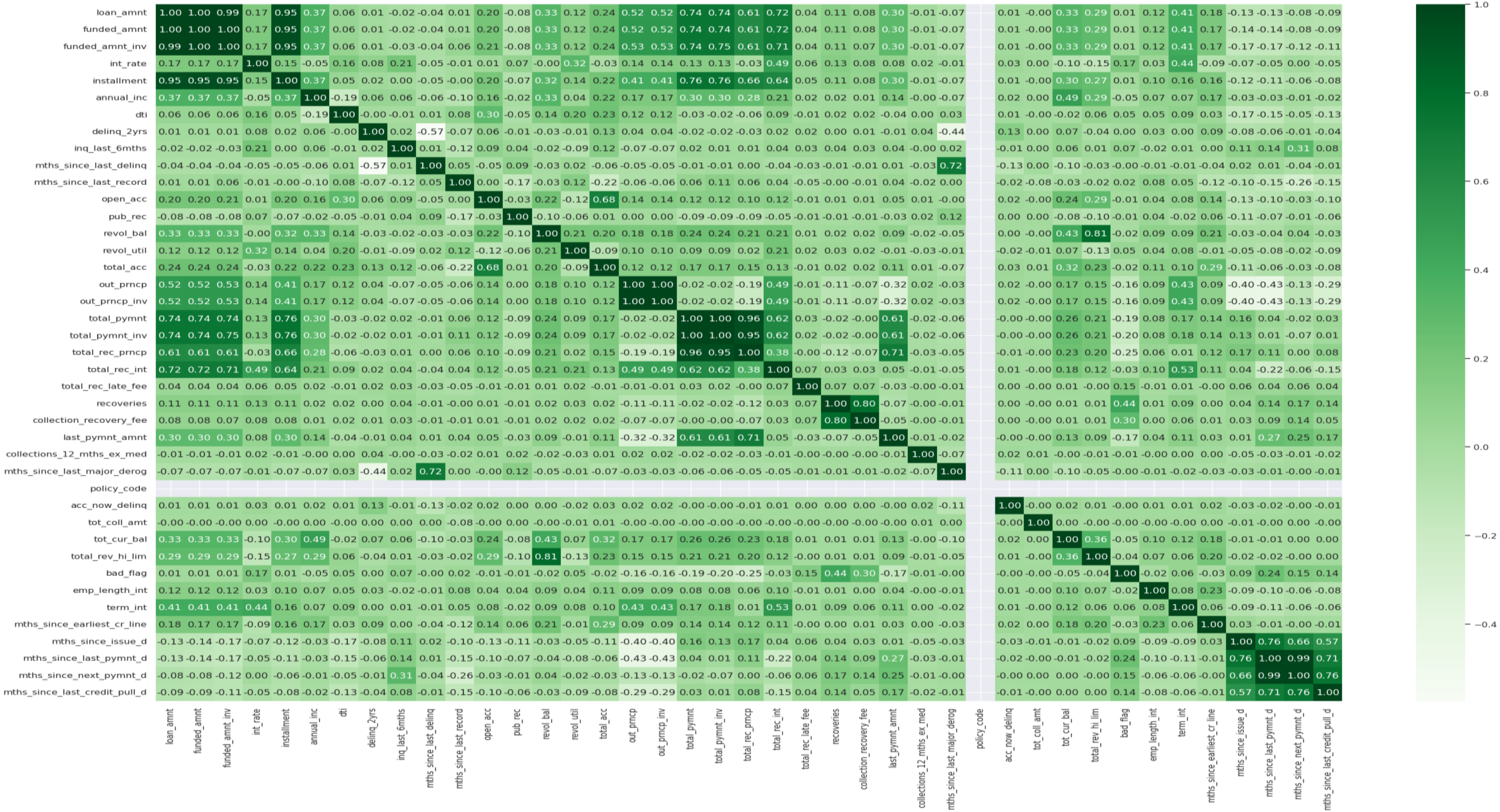
bad_flag
0 means good
1 means bad



90%

10%

Correlation Check



03

Missing Values

Checking Missing Values

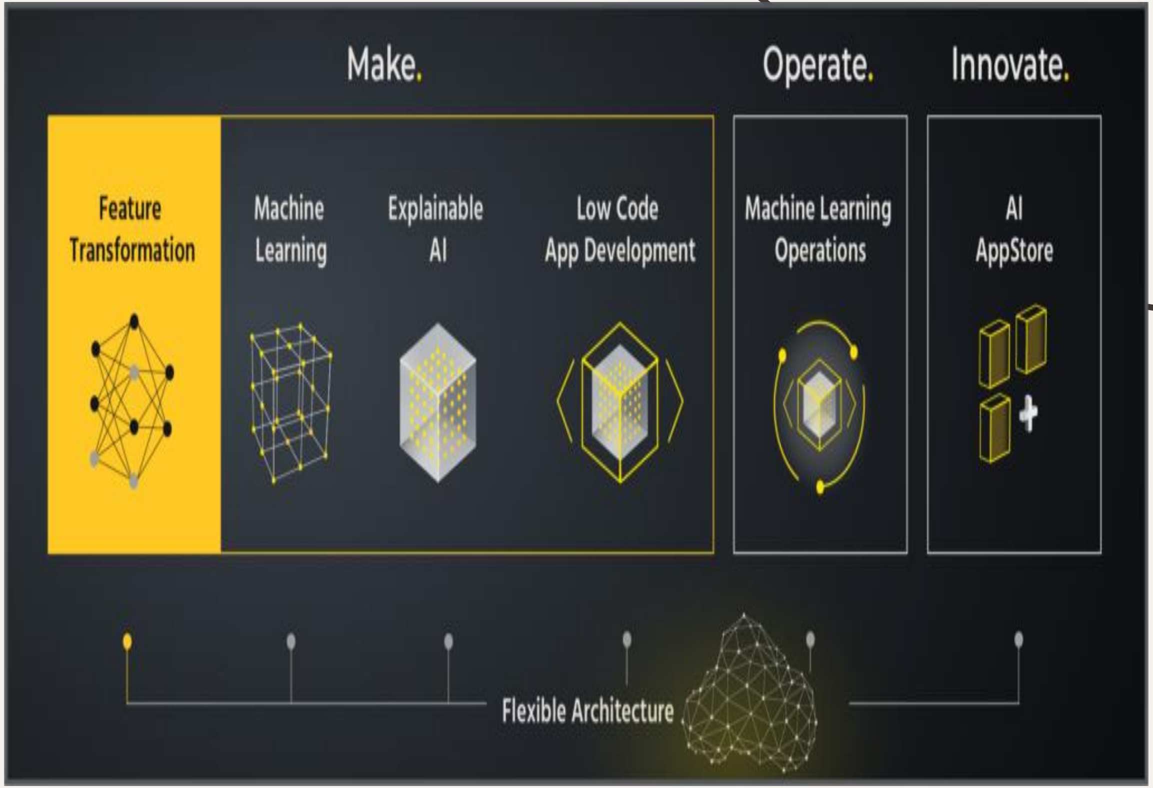
	Total Null Values	Percentage	Data Type	NULL Values
mths_since_last_record	403647	86.566585	float64	403647
mths_since_last_delinq	250351	53.690554	float64	250351
tot_coll_amt	70276	15.071469	float64	70276
tot_cur_bal	70276	15.071469	float64	70276
emp_length_int	21008	4.505399	int64	21008
revol_util	340	0.072917	float64	340
collections_12_mths_ex_med	145	0.031097	float64	145
total_acc	29	0.006219	float64	29
pub_rec	29	0.006219	object	29
open_acc	29	0.006219	float64	29
acc_now_delinq	29	0.006219	float64	29
inq_last_6mths	29	0.006219	float64	29
mths_since_earliest_cr_line	29	0.006219	float64	29
delinq_2yrs	29	0.006219	float64	29
annual_inc	4	0.000858	float64	4

Missing Values Filling

```
data['annual_inc'].fillna(data['annual_inc'].mean(), inplace=True)
data['mths_since_earliest_cr_line'].fillna(0, inplace=True)
data['acc_now_delinq'].fillna(0, inplace=True)
data['total_acc'].fillna(0, inplace=True)
data['pub_rec'].fillna(0, inplace=True)
data['open_acc'].fillna(0, inplace=True)
data['inq_last_6mths'].fillna(0, inplace=True)
data['delinq_2yrs'].fillna(0, inplace=True)
data['collections_12_mths_ex_med'].fillna(0, inplace=True)
data['revol_util'].fillna(0, inplace=True)
data['emp_length_int'].fillna(0, inplace=True)
data['tot_cur_bal'].fillna(0, inplace=True)
data['tot_coll_amt'].fillna(0, inplace=True)
data['mths_since_last_delinq'].fillna(-1, inplace=True)
```

04

Feature Scaling and Transformation



For categorical data types, one hot encoding method will be used.

One-hot encoding in machine learning is the conversion of categorical information into a format that may be fed into machine learning algorithms to improve prediction accuracy.

id	color
1	red
2	blue
3	green
4	blue

One Hot Encoding

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Next will be shown what the result of the one hot encoding method look like

	grade_B	grade_C	grade_D	grade_E	grade_F	grade_G	home_ownership_MORTGAGE	home_ownership_NONE	home_ownership_OTHER
0	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0

purpose_renewable_energy	purpose_small_business	purpose_vacation	purpose_wedding	addr_state_AL	addr_state_AR	addr_state_AZ	addr_state_CA
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0

home_ownership_OWN	home_ownership_RENT	verification_status_Source Verified	verification_status_Verified	purpose_credit_card	purpose_debt_consolidation
0	1	0	1	1	0
0	1	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
0	1	1	0	0	0

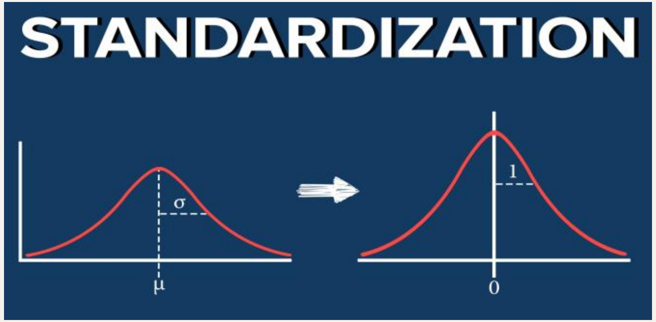
addr_state_NV	addr_state_NY	addr_state_OH	addr_state_OK	addr_state_OR	addr_state_PA	addr_state_RI	addr_state_SC	addr_state_SD
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0

purpose_educational	purpose_home_improvement	purpose_house	purpose_major_purchase	purpose_medical	purpose_moving	purpose_other
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	1

addr_state_TX	addr_state_UT	addr_state_VA	addr_state_VT	addr_state_WA	addr_state_WI	addr_state_WV	addr_state_WY	initial_list_status_w
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

For categorical data types, standardization method will be used.

Standardization entails scaling data to fit a standard normal distribution. A standard normal distribution is defined as a distribution with a mean of 0 and a standard deviation of 1.



Next will be shown what the result of the standardization method look like

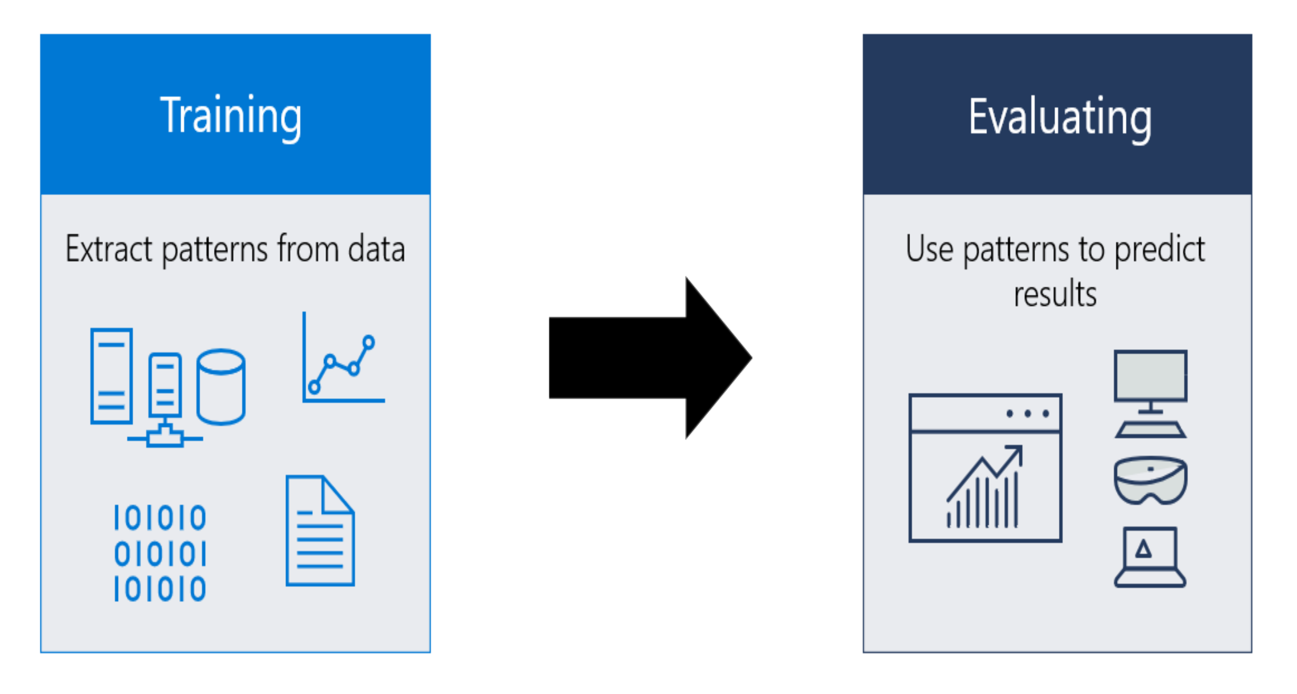
index	loan_amnt	int_rate	annual_inc	dti	delinq_2yrs	inq_last_6mths	mths_since_last_delinq
0	-1.124392356712422	-0.7295871896779282	-0.8965510588470832	1.328632303633231	-0.3570117327421637	0.1789197184672106	-0.7087922647233157
1	-1.4260878754457609	0.33063384473679663	-0.7873872560523542	-2.06579097253051	-0.3570117327421637	3.843327893698174	-0.7087922647233157
2	-1.4381556961950943	0.4889785446818533	-1.1102937847191625	-1.082490871517974	-0.3570117327421637	1.0950217622749514	-0.7087922647233157
3	-0.5210013192457448	-0.07784958410697172	-0.43806308710922126	0.3542481361790991	-0.3570117327421637	0.17891971846721064	0.8608112361730155
4	-1.365748771699093	-0.2614376420142836	0.12231110057038769	0.09186494860321535	-0.3570117327421637	-0.7371823253405302	0.9916115279143765

open_acc	pub_rec	revol_bal	revol_util	total_acc	out_prncp	total_rec_late_fee	recoveries
-1.6411655374641578	-0.3142896504323558	-0.12488758171695738	1.159498371158884	-1.3845570216026701	-0.6939437309577768	-0.1234634709371513	-0.1545487451411822
-1.6411655374641578	-0.3142896504323558	-0.7033781508596487	-1.9659798658018848	-1.8155375684221375	-0.6939437309577768	-0.1234634709371513	0.0574699356338950
-1.8416408284409005	-0.3142896504323558	-0.6420033049235168	1.782070079194138	-1.2983609122387767	-0.6939437309577768	-0.1234634709371513	-0.1545487451411822
-0.23783850062696055	-0.3142896504323558	-0.5142236808389928	-1.4780182568012803	1.0289340405863474	-0.6939437309577768	3.0992643538484064	-0.1545487451411822
0.7645379542567519	-0.3142896504323558	0.5587479165550886	-0.09405817610128982	1.1151301499502408	-0.5732684647254045	-0.1234634709371513	-0.1545487451411822

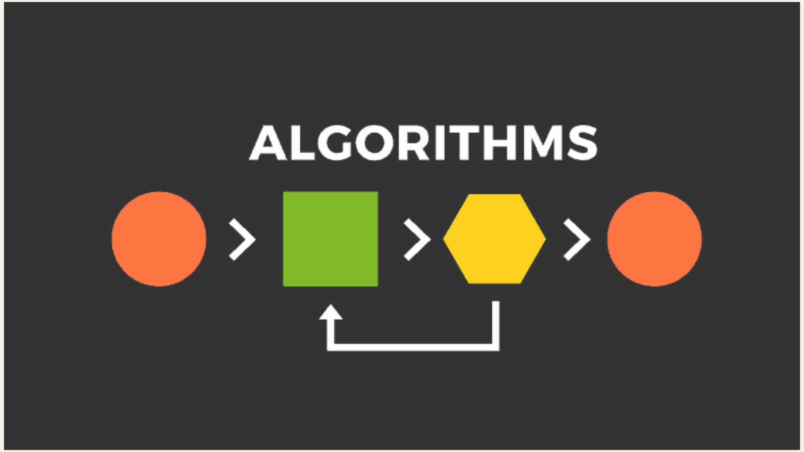
collections_12_mths_ex_med	acc_now_delinq	tot_coll_amt	tot_cur_bal	emp_length_int
-0.08360769477500803	-0.05830651637302818	-0.012088616232763732	-0.7926483414923741	1.1386054448594471
-0.08360769477500803	-0.05830651637302818	-0.012088616232763732	-0.7926483414923741	-1.523744478548263
-0.08360769477500803	-0.05830651637302818	-0.012088616232763732	-0.7926483414923741	1.1386054448594471
-0.08360769477500803	-0.05830651637302818	-0.012088616232763732	-0.7926483414923741	1.1386054448594471
-0.08360769477500803	-0.05830651637302818	-0.012088616232763732	-0.7926483414923741	-1.257509486207492

05

Modeling


Training
Extract patterns from data
101010
010101
101010
Evaluating
Use patterns to predict results

Algorithms



Decision Tree

Naive Bayes

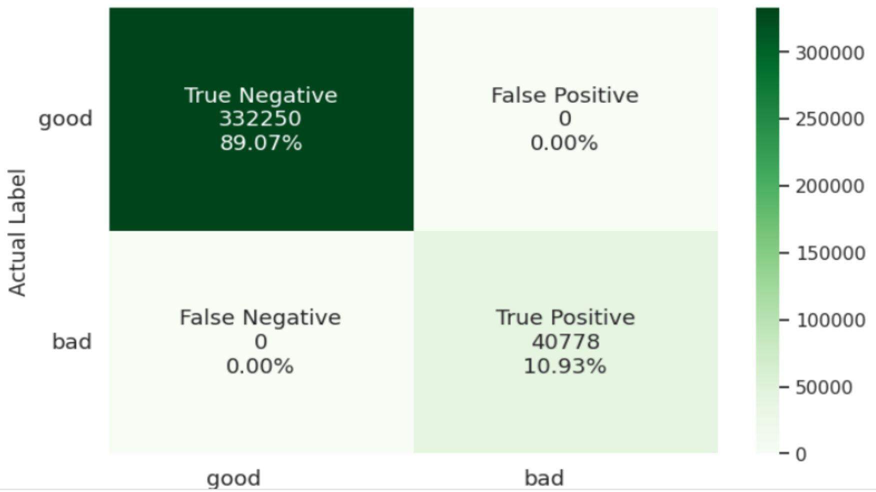
Adaboost

Decision Tree

	Evaluation Metrics	Train	Test	Diff Range
0	Accuracy	1.000000	0.903000	0.097000
1	Precision	1.000000	0.555000	0.445000
2	Recall	1.000000	0.583000	0.417000
3	F1 Score	1.000000	0.569000	0.431000
4	F1 Score (crossval)	1.000000	0.571000	0.429000
5	ROC AUC	1.000000	0.763000	0.237000
6	ROC AUC (crossval)	1.000000	0.765000	0.235000

```
==== Actual Data (Train) =====
Total = 373028
good = 332250
bad = 40778
==== Predicted Data (Train) =====
TP = 40778, FP = 0, TN = 332250, FN = 0
Predictly Correct = 373028
Predictly Wrong = 0
```

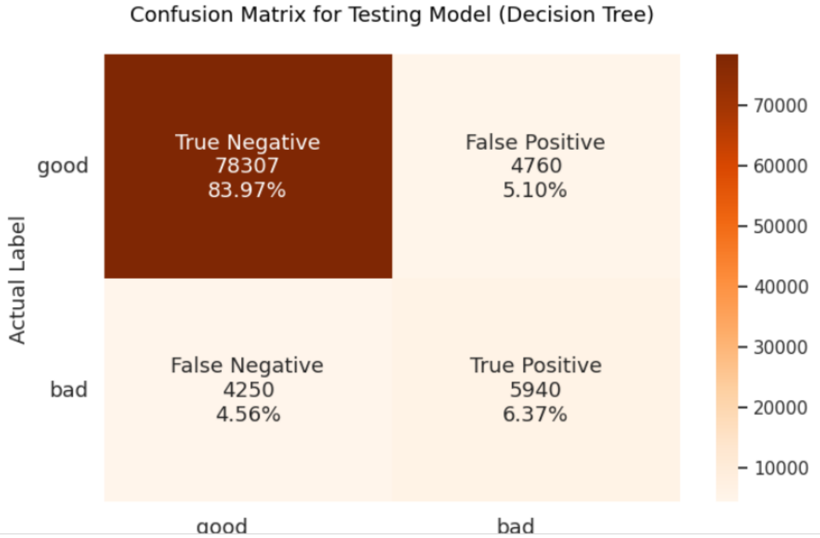
Confusion Matrix for Training Model (Decision Tree)



Decision Tree

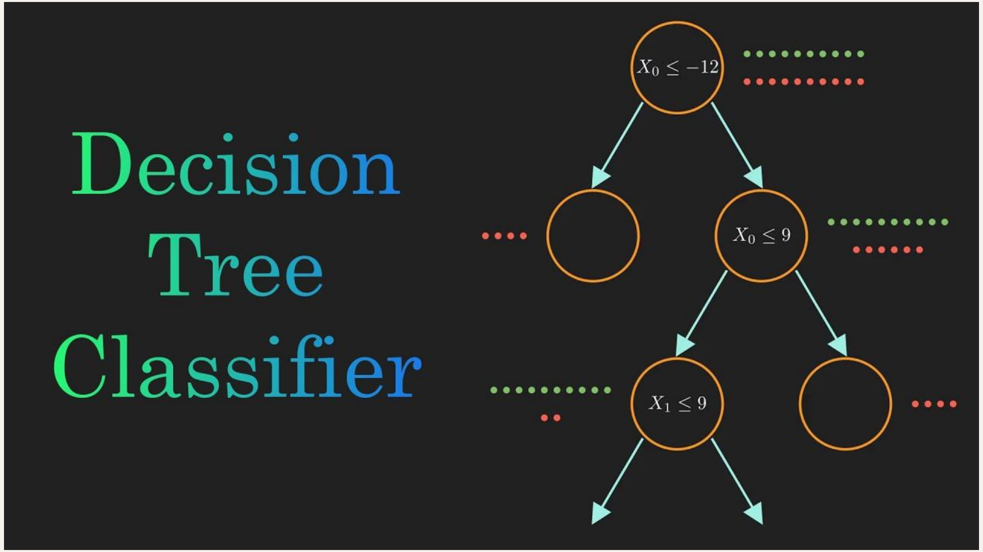
	Evaluation Metrics	Train	Test	Diff Range
0	Accuracy	1.000000	0.903000	0.097000
1	Precision	1.000000	0.555000	0.445000
2	Recall	1.000000	0.583000	0.417000
3	F1 Score	1.000000	0.569000	0.431000
4	F1 Score (crossval)	1.000000	0.571000	0.429000
5	ROC AUC	1.000000	0.763000	0.237000
6	ROC AUC (crossval)	1.000000	0.765000	0.235000

==== Actual Data (Test) =====
Total = 93257
good = 83067
bad = 10190
==== Predicted Data (Test) =====
TP = 5940, FP = 4760, TN = 78307, FN = 4250
Predictly Correct = 84247
Predictly Wrong = 9010



Decision Tree

Training Accuracy: 100.0 %
Testing Accuracy: 90.34 %

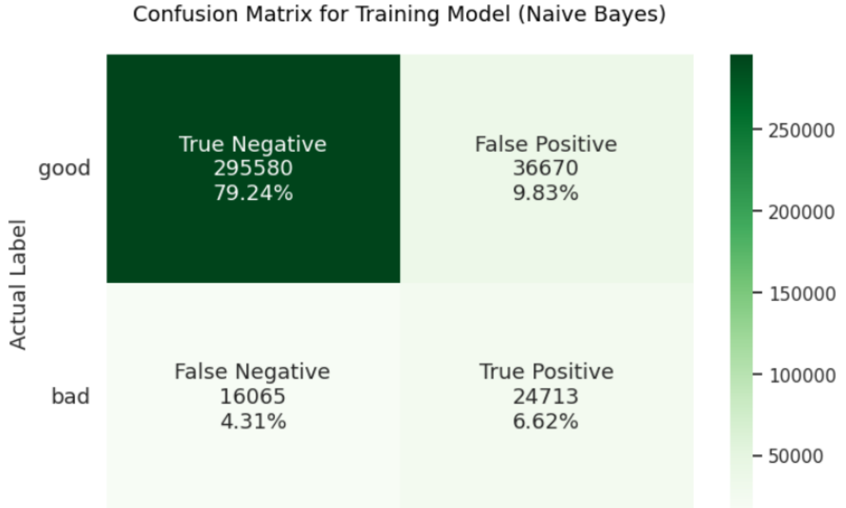


Precision	Recall	F-Score
Train = 1.0 Test = 0.55	Train = 1.0 Test = 0.58	Train = 1.0 Test = 0.56

Naive Bayes

	Evaluation Metrics	Train	Test	Diff Range
0	Accuracy	0.859000	0.859000	0.000000
1	Precision	0.403000	0.404000	-0.001000
2	Recall	0.606000	0.610000	-0.004000
3	F1 Score	0.484000	0.486000	-0.002000
4	F1 Score (crossval)	0.529000	0.529000	0.000000
5	ROC AUC	0.811000	0.810000	0.001000
6	ROC AUC (crossval)	0.810000	0.809000	0.001000

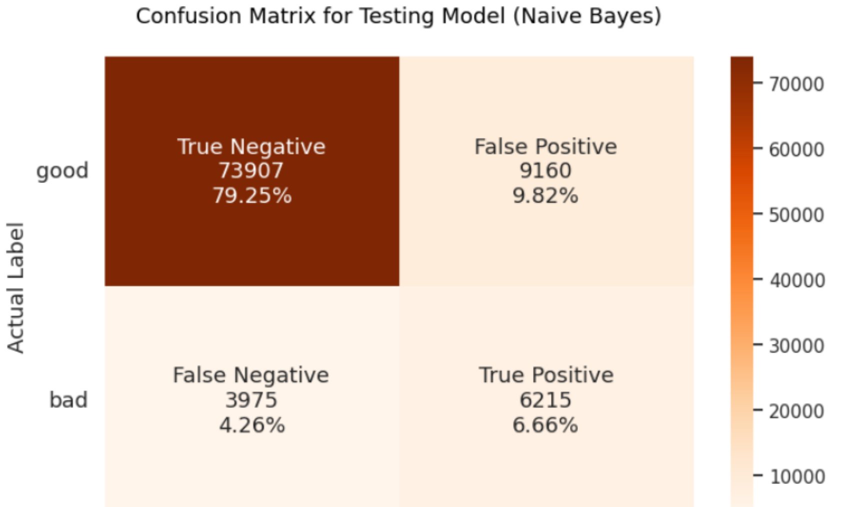
```
==== Actual Data (Train) =====
Total = 373028
good = 332250
bad = 40778
==== Predicted Data (Train) =====
TP = 24713, FP = 36670, TN = 295580, FN = 16065
Predictly Correct = 320293
Predictly Wrong = 52735
```



Naive Bayes

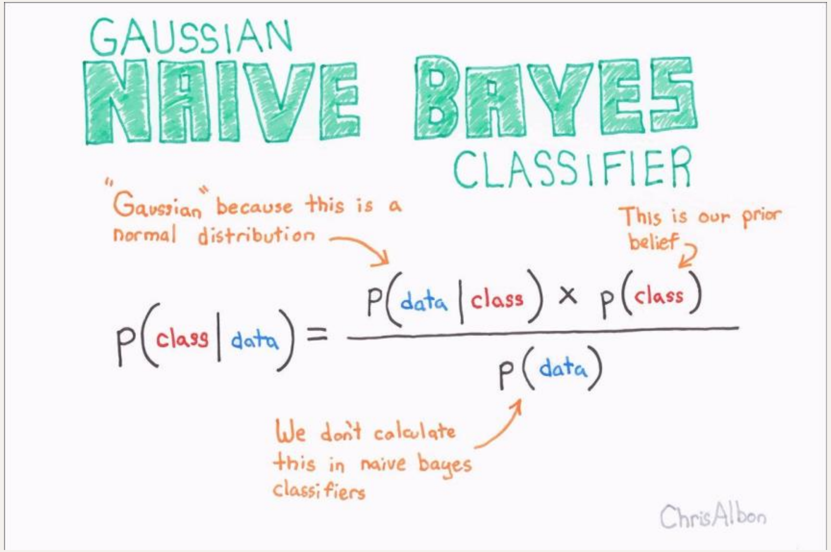
	Evaluation Metrics	Train	Test	Diff Range
0	Accuracy	0.859000	0.859000	0.000000
1	Precision	0.403000	0.404000	-0.001000
2	Recall	0.606000	0.610000	-0.004000
3	F1 Score	0.484000	0.486000	-0.002000
4	F1 Score (crossval)	0.529000	0.529000	0.000000
5	ROC AUC	0.811000	0.810000	0.001000
6	ROC AUC (crossval)	0.810000	0.809000	0.001000

```
==== Actual Data (Test) =====
Total = 93257
good = 83067
bad = 10190
==== Predicted Data (Test) =====
TP = 6215, FP = 9160, TN = 73907, FN = 3975
Predictly Correct = 80122
Predictly Wrong = 13135
```



Naive Bayes

Training Accuracy: 85.86 %
Test Accuracy: 85.92 %



Precision

Train = 0.403
Test = 0.404

Recall

Train = 0.60
Test = 0.61

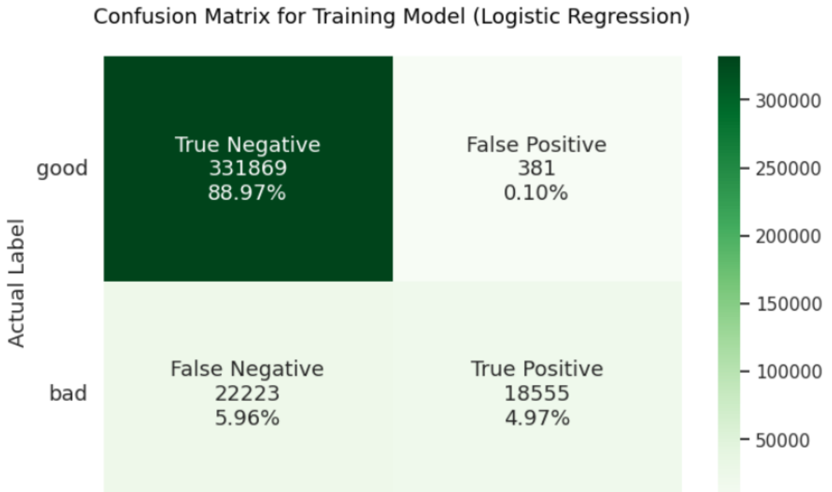
F-Score

Train = 0.484
Test = 0.486

Logistic Regression

	Evaluation Metrics	Train	Test	Diff Range
0	Accuracy	0.939000	0.940000	-0.001000
1	Precision	0.980000	0.981000	-0.001000
2	Recall	0.455000	0.456000	-0.001000
3	F1 Score	0.621000	0.623000	-0.002000
4	F1 Score (crossval)	0.623000	0.623000	0.000000
5	ROC AUC	0.855000	0.856000	-0.001000
6	ROC AUC (crossval)	0.856000	0.855000	0.001000

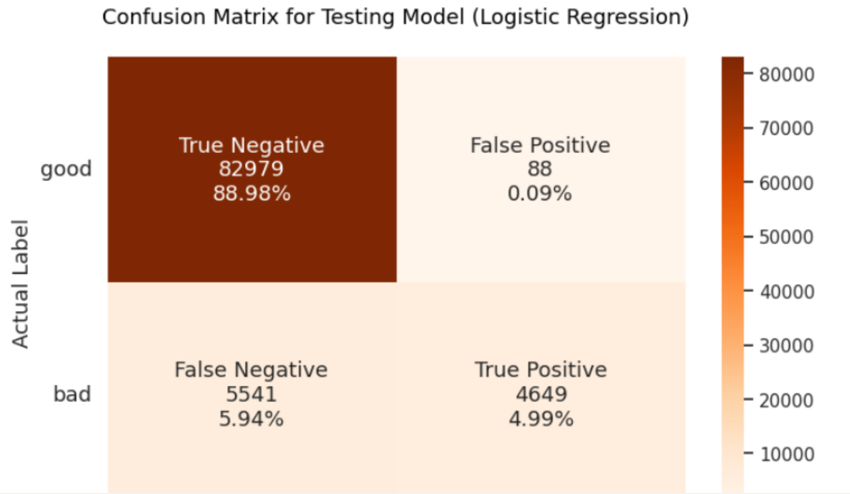
```
==== Actual Data (Train) =====
Total = 373028
good = 332250
bad = 40778
==== Predicted Data (Train) =====
TP = 18555, FP = 381, TN = 331869, FN = 22223
Predictly Correct = 350424
Predictly Wrong = 22604
```



Logistic Regression

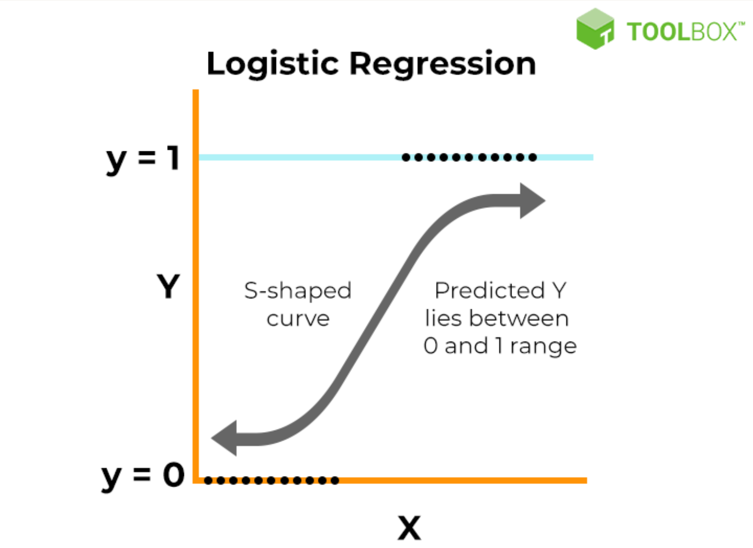
	Evaluation Metrics	Train	Test	Diff Range
0	Accuracy	0.939000	0.940000	-0.001000
1	Precision	0.980000	0.981000	-0.001000
2	Recall	0.455000	0.456000	-0.001000
3	F1 Score	0.621000	0.623000	-0.002000
4	F1 Score (crossval)	0.623000	0.623000	0.000000
5	ROC AUC	0.855000	0.856000	-0.001000
6	ROC AUC (crossval)	0.856000	0.855000	0.001000

==== Actual Data (Test) =====
Total = 93257
good = 83067
bad = 10190
==== Predicted Data (Test) =====
TP = 4649, FP = 88, TN = 82979, FN = 5541
Predictly Correct = 87628
Predictly Wrong = 5629



Logistic Regression

Training Accuracy: 93.94 %
Test Accuracy: 93.96 %



Precision

Train = 0.980
Test = 0.981

Recall

Train = 0.455
Test = 0.456

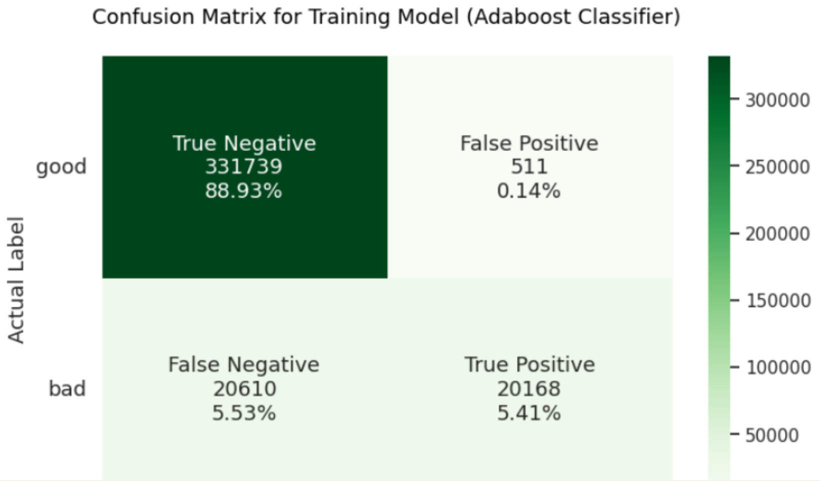
F-Score

Train = 0.621
Test = 0.623

Adaboost

	Evaluation Metrics	Train	Test	Diff Range
0	Accuracy	0.943000	0.944000	-0.001000
1	Precision	0.975000	0.982000	-0.007000
2	Recall	0.495000	0.496000	-0.001000
3	F1 Score	0.656000	0.659000	-0.003000
4	F1 Score (crossval)	0.657000	0.657000	0.000000
5	ROC AUC	0.876000	0.874000	0.002000
6	ROC AUC (crossval)	0.876000	0.875000	0.001000

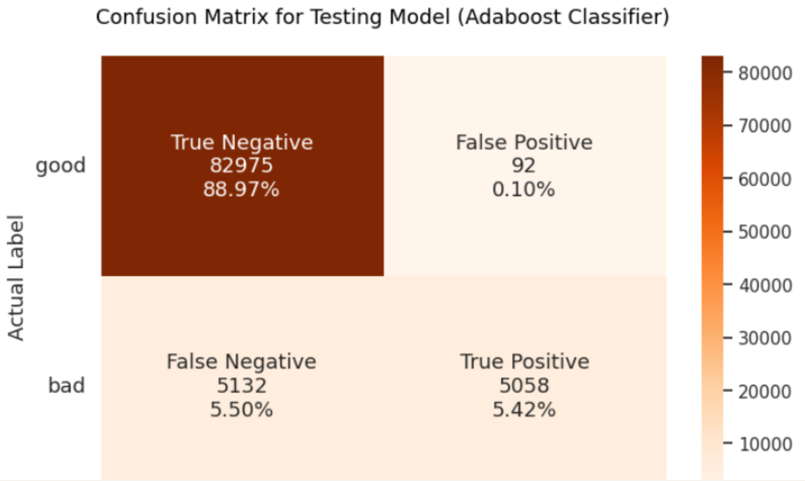
```
==== Actual Data (Train) =====
Total = 373028
good = 332250
bad = 40778
==== Predicted Data (Train) =====
TP = 20168, FP = 511, TN = 331739, FN = 20610
Predictly Correct = 351907
Predictly Wrong = 21121
```



Adaboost

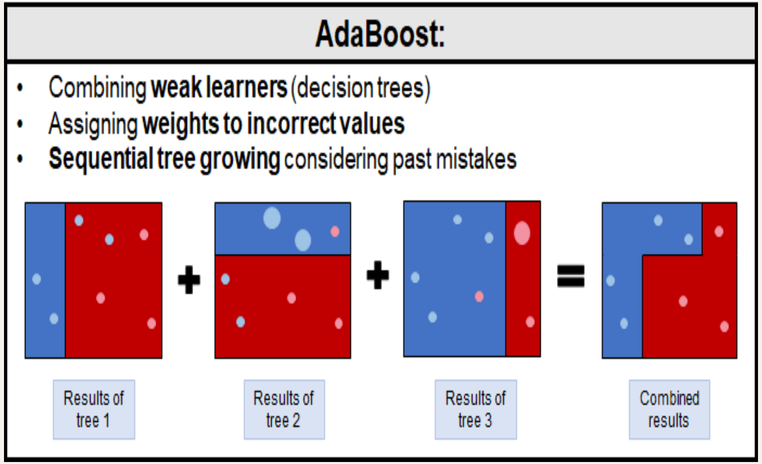
	Evaluation Metrics	Train	Test	Diff Range
0	Accuracy	0.943000	0.944000	-0.001000
1	Precision	0.975000	0.982000	-0.007000
2	Recall	0.495000	0.496000	-0.001000
3	F1 Score	0.656000	0.659000	-0.003000
4	F1 Score (crossval)	0.657000	0.657000	0.000000
5	ROC AUC	0.876000	0.874000	0.002000
6	ROC AUC (crossval)	0.876000	0.875000	0.001000

==== Actual Data (Test) =====
Total = 93257
good = 83067
bad = 10190
==== Predicted Data (Test) =====
TP = 5058, FP = 92, TN = 82975, FN = 5132
Predictly Correct = 88033
Predictly Wrong = 5224



Adaboost

Training Accuracy: 94.34 %
Test Accuracy: 94.4 %



Precision

Train = 0.97
Test = 0.98

Recall

Train = 0.495
Test = 0.496

F-Score

Train = 0.656
Test = 0.659

05

Conclusion

	Models	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	F1 Score (Train)	F1 Score (Test)
0	Adaboost Classifier	0.975000	0.982000	0.495000	0.496000	0.656000	0.659000
1	Logistic Regression	0.980000	0.981000	0.455000	0.456000	0.621000	0.623000
2	Decision Tree	1.000000	0.555000	1.000000	0.583000	1.000000	0.569000
3	Naive Bayes	0.403000	0.404000	0.606000	0.610000	0.484000	0.486000

Thanks