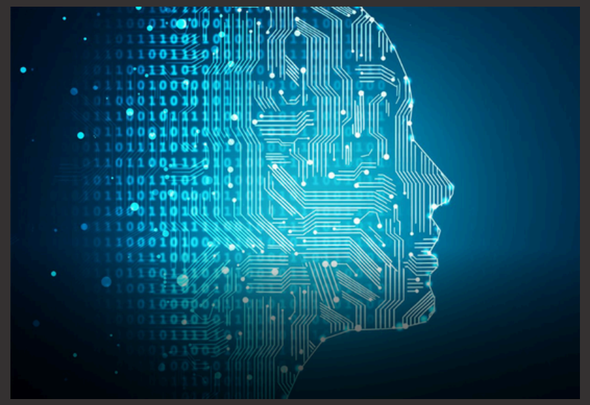
Credit Risk Model

Au Hong Yao Justin U2310037F
Lee Yew Joe U2322894G

FCED Team 7

Table Of Contents

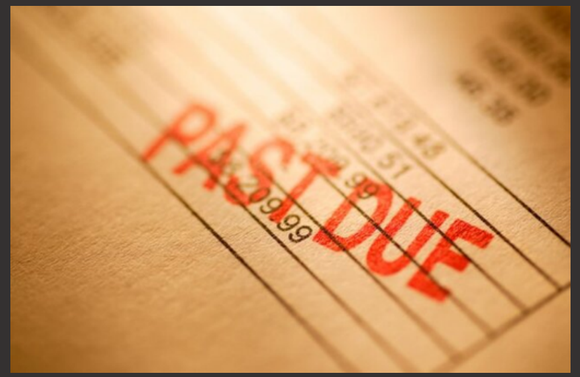
Motivation & Problem Statement



Problem Definition

How can we enhance the current credit risk models available such that the rate of payments defaulting decreases significantly?

What are some of the **contributing factors** to this risk levels of loaners?



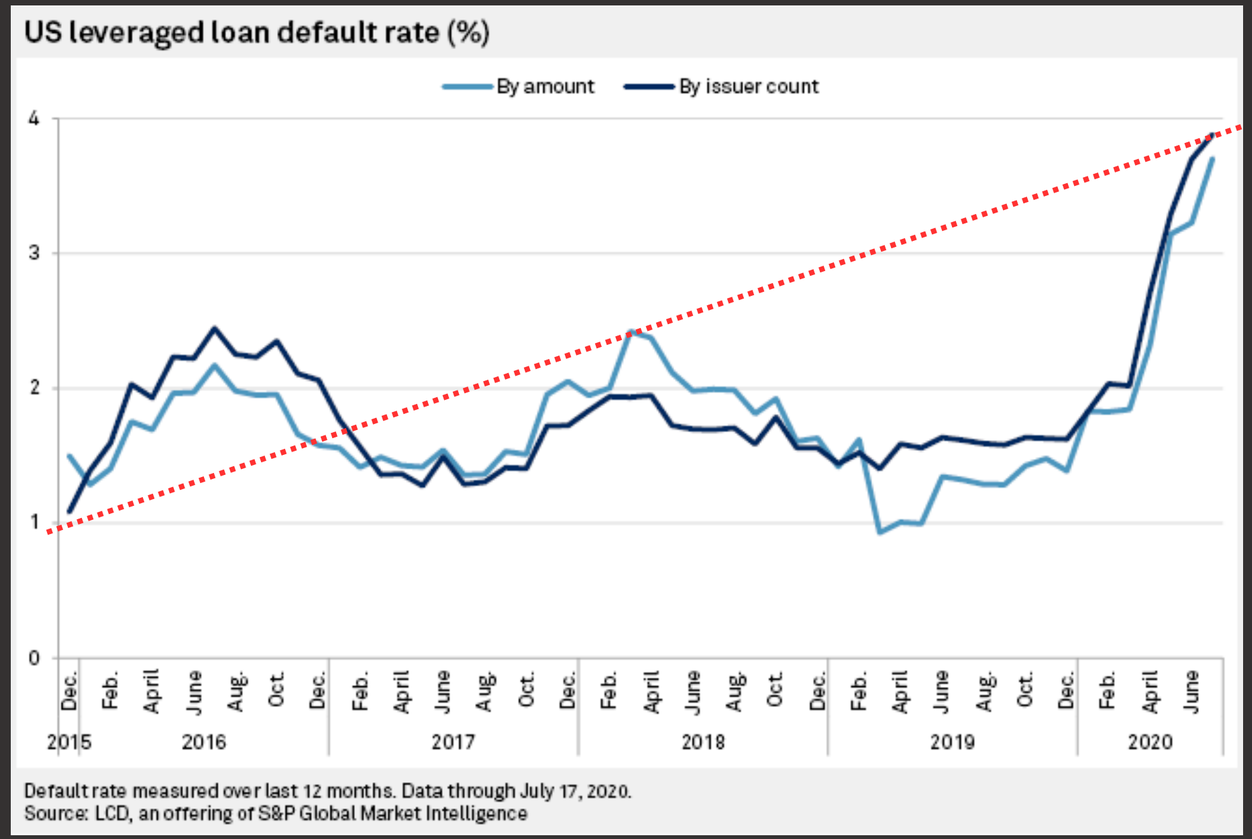
Motivation

Loans are increasingly more common. However, there has been an significant increase in loan defaults too.

Motivation

Rate at which loan default are increasing poses a significant issue to banks and the ecnonomy itself

Aim is to refine the models available such that loan takers risk are carefully analysed before approving their loan



Loan Dataset

Variables

- Loan Amount
- Interest Rate
- Monthly Income
- ...**many more!**

Data Points

- 421,094 Rows
- 74 Columns

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	...	total_bal_il	il_
0	60516983	64537751	20000	20000	20000	36 months	12.29	667.06	C	C1	...	NaN	
1	60187139	64163931	11000	11000	11000	36 months	12.69	369.00	C	C2	...	NaN	
2	60356453	64333218	7000	7000	7000	36 months	9.99	225.84	B	B3	...	NaN	
3	59955769	63900496	10000	10000	10000	36 months	10.99	327.34	B	B4	...	NaN	
4	58703693	62544456	9550	9550	9550	36 months	19.99	354.87	E	E4	...	NaN	

5 rows × 74 columns

2: Data Preparations

1. Dropping rows that are null
2. Re-Categorize values in ‘loan_status’
3. Dropping columns

Data Preparation

Dropping these columns as they have too much empty rows

```
# drop columns with more than 40% null values
loandata.dropna(thresh = loandata.shape[0]*0.9, axis = 1, inplace = True)
```

```
#printing out a score based on number of null value in each columm that has more than 40% of null values
na_values = loandata.isnull().mean()
na_values[na_values>0.4]
```

```
desc                          0.999893
mths_since_last_delinq        0.484360
mths_since_last_record        0.823282
mths_since_last_major_derog   0.708547
annual_inc_joint              0.998786
dti_joint                     0.998791
verification_status_joint     0.998786
open_acc_6m                   0.949246
open_il_6m                    0.949246
open_il_12m                   0.949246
open_il_24m                   0.949246
mths_since_rcnt_il            0.950581
total_bal_il                  0.949246
il_util                       0.955789
open_rv_12m                   0.949246
open_rv_24m                   0.949246
max_bal_bc                    0.949246
all_util                      0.949246
inq_fi                        0.949246
total_cu_tl                   0.949246
inq_last_12m                  0.949246
dtype: float64
```

Data Preparation

```
loandata['loan_status'].value_counts()
```

```
loan_status
Current                 377553
Fully Paid               22984
Issued                    8460
Late (31-120 days)        4691
In Grace Period           3107
Charged Off               2773
Late (16-30 days)         1139
Default                    387
Name: count, dtype: int64
```

```
# Dropping columns that will not help us in predicting the credit risk

indexnew = loandata[ (loandata['loan_status'] == 'Current') ].index
loandata.drop(indexnew , inplace=True)
indexnew = loandata[ (loandata['loan_status'] == 'Issued')  ].index
loandata.drop(indexnew , inplace=True)
indexnew = loandata[ (loandata['loan_status'] == 'In Grace Period') ].index
loandata.drop(indexnew , inplace=True)
```

```
# Inserting column 'credit_risk' to categorise each row into 'Risky' or 'Non-Risky'
loandata.insert(0, 'credit_risk', [0 if status == 'Fully Paid' else 1 for status in loandata['loan_status']])
loandata['credit_risk'].value_counts()
```

```
credit_risk
0    22984
1     8990
Name: count, dtype: int64
```

We decided to drop rows such as “Current/In Grace Period /Issued” and re categorize the remaining into **risk** and **non-risky** under **“credit_risk”**

2: EDA

1. Chi Square Test
2. Plotting and Pattern observation of graphs

EDA

Based on the Chi Square Test, we decided to drop variable with chi score less than 100 as they do not make good predictors for our response variable

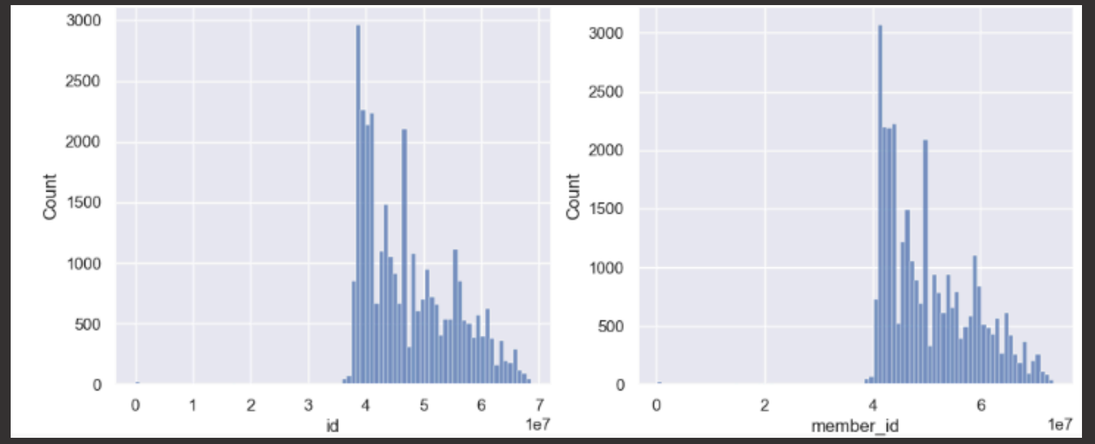
All features with their scores sorted in ascending order:

	Feature	Scores
11	title	0.013406
19	application_type	2.205872
7	issue_d	29.564775
10	purpose	35.594692
14	earliest_cr_line	54.494849
15	initial_list_status	58.808054
13	addr_state	62.777998
6	verification_status	106.798471
4	emp_length	253.203447
0	term	287.352533
5	home_ownership	346.806507
16	last_pymnt_d	388.585085
18	last_credit_pull_d	1109.595160
1	grade	1803.404755
12	zip_code	4615.609560
17	next_pymnt_d	6579.315922
2	sub_grade	7633.277385
3	emp_title	313821.899639
9	url	568542.924782
8	pymnt_plan	NaN

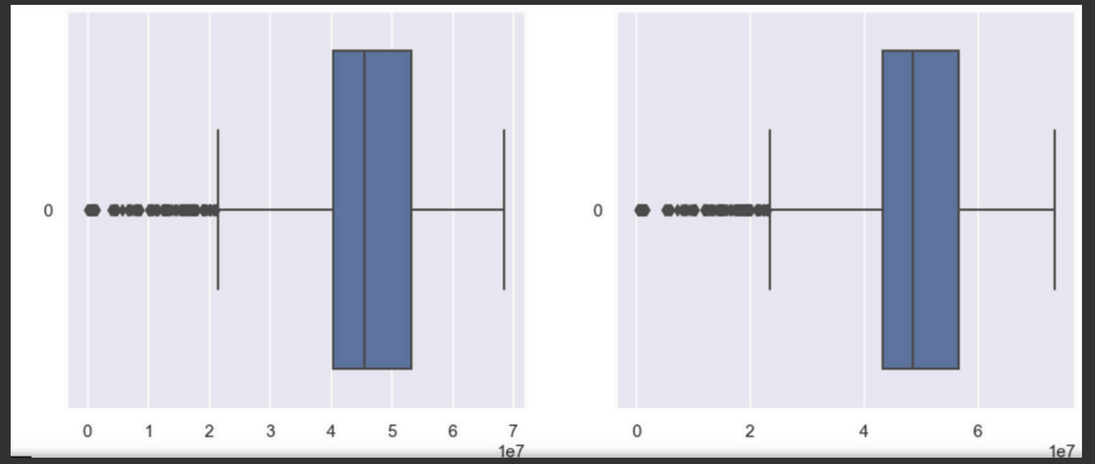
```
#based on chi test we chose the following to drop for chi square score under 100
loandata.drop(columns = ['title', 'application_type', 'issue_d' ,'purpose', 'earliest_cr_line', 'initial_list_status', 'addr_state'], inplace = True)
```

EDA

Plotting the distribution of numeric variables to observe any patterns (Boxplot and Histogram)



Insights: Observed that a few variable has extremely skewed distribuition which might cause biasness in the model. Hence, we drop them



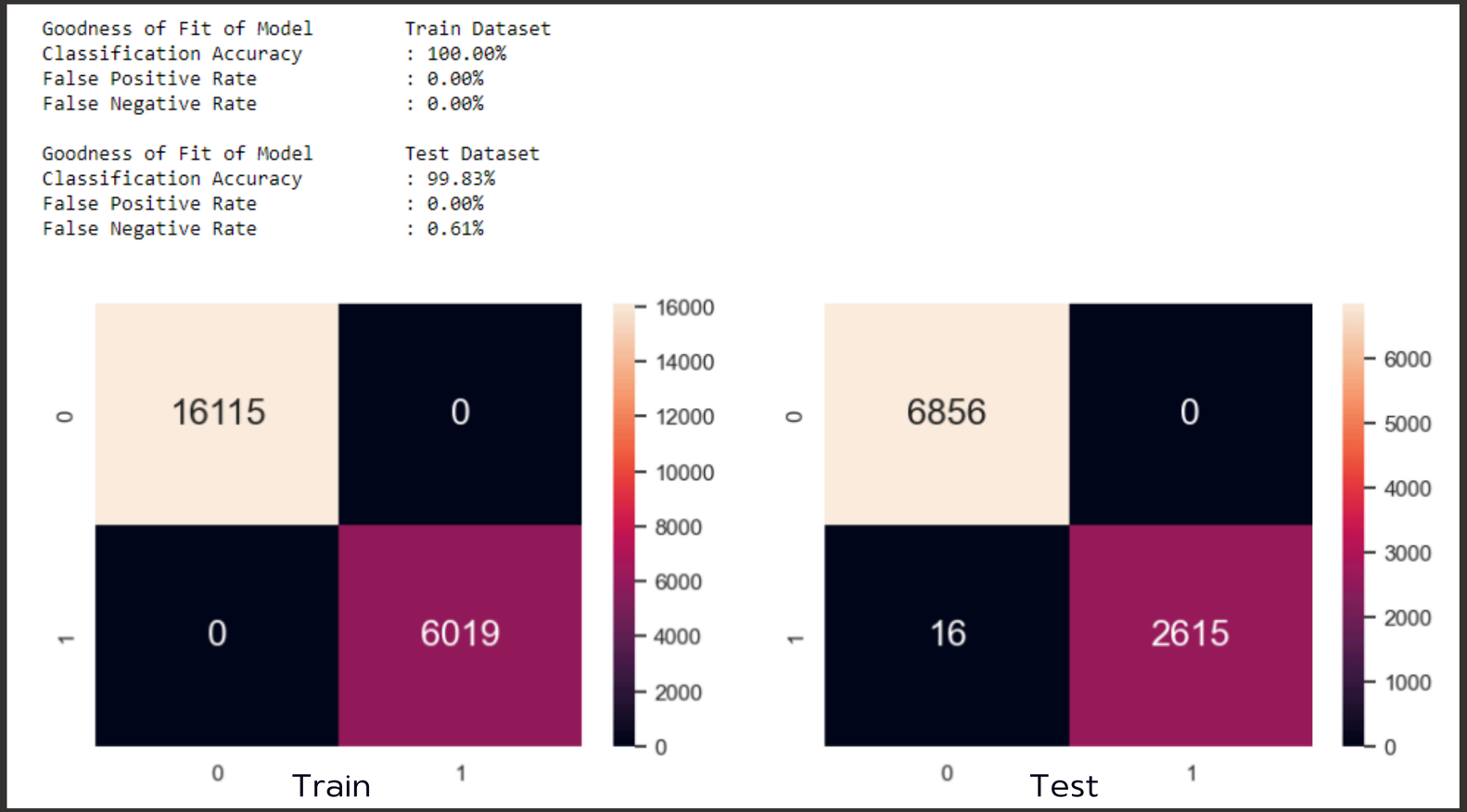
Machine Learning

1. Random Forest 
2. Logistic Regression 
3. Artificial Neural Network 

Random Forest

Random forests aim to reduce the overfitting problem of decision trees by averaging multiple trees, improving accuracy and robustness.

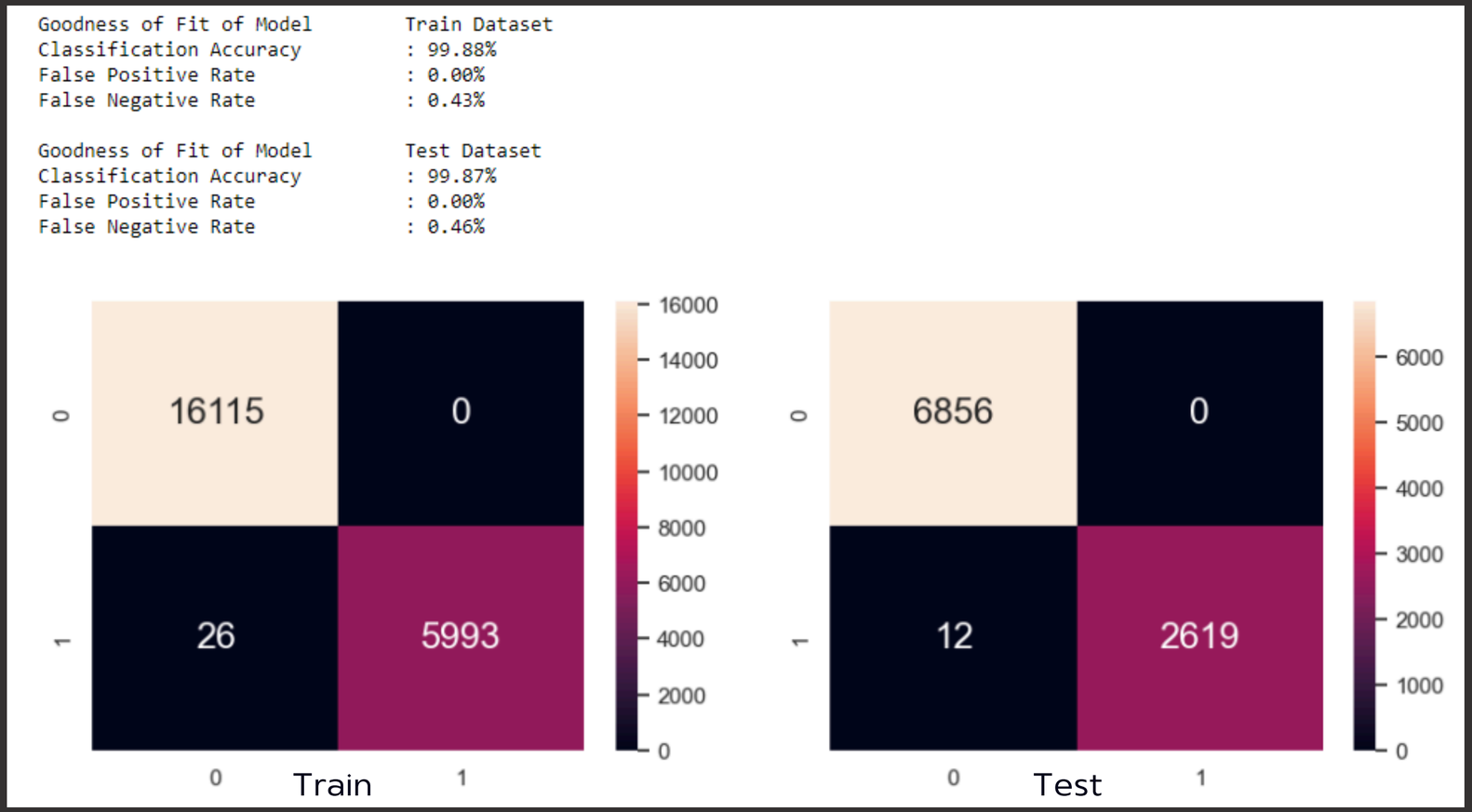
```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 100.00%
False Positive Rate             : 0.00%
False Negative Rate             : 0.00%

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 99.83%
False Positive Rate             : 0.00%
False Negative Rate             : 0.61%
```



Logistic Regression

Logistic regression is a supervised machine learning algorithm that uses a logistic function for binary classification tasks

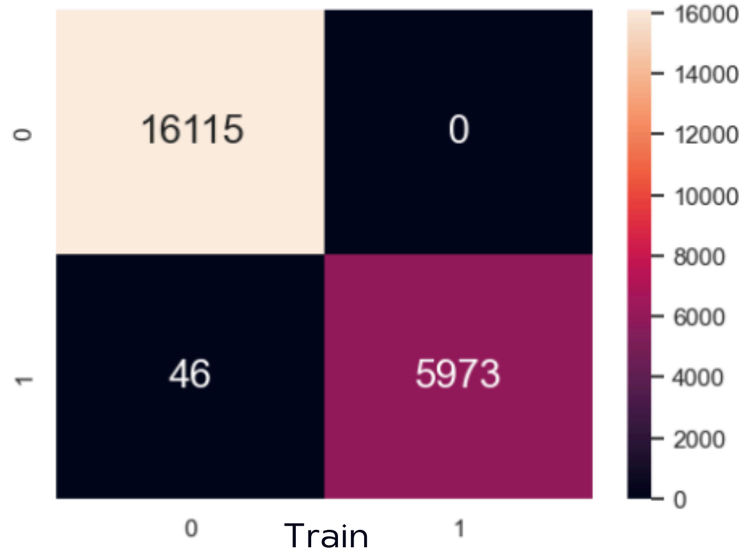
```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 99.88%
False Positive Rate             : 0.00%
False Negative Rate             : 0.43%

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 99.87%
False Positive Rate             : 0.00%
False Negative Rate             : 0.46%
```

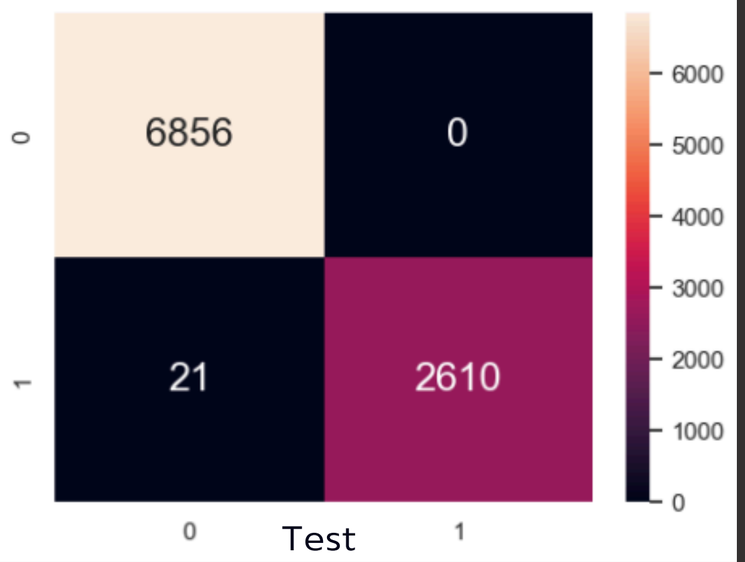


Artificial Neural Network

Neural network is a machine learning model inspired by the structure and function of biological neural networks in animal brains

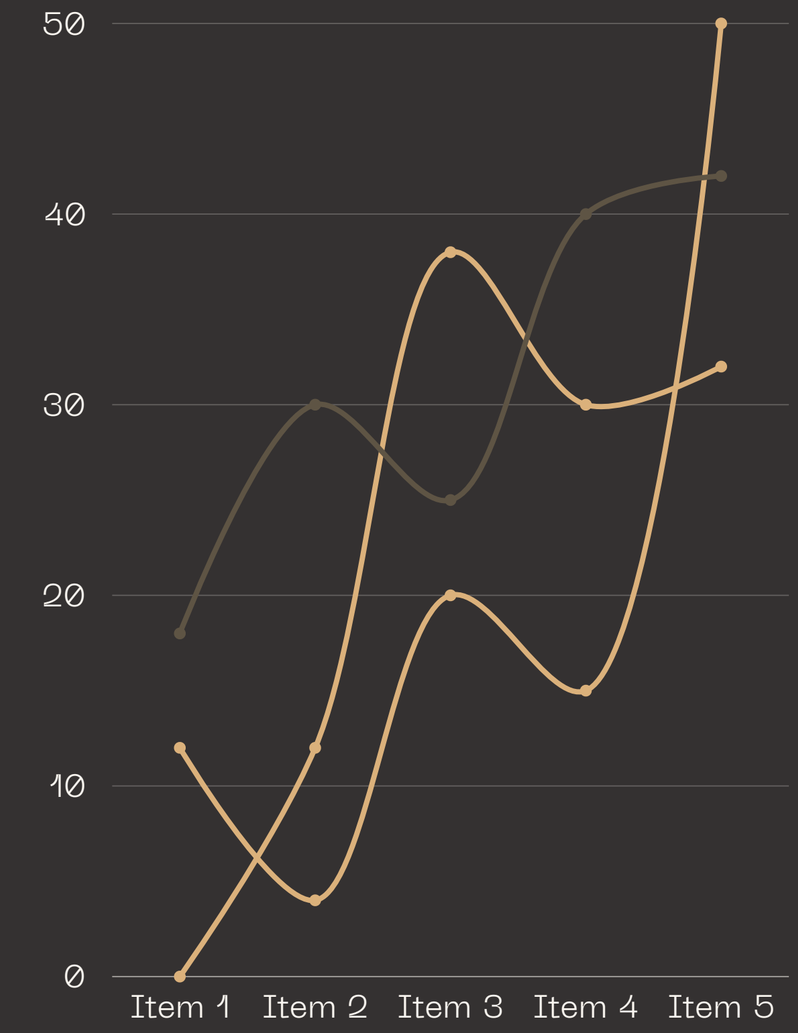
```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 99.79%
False Positive Rate             : 0.00%
False Negative Rate             : 0.76%

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 99.78%
False Positive Rate             : 0.00%
False Negative Rate             : 0.80%
```





Refining the models

Removing outliers from the dataset



Results after removing outliers

```
Goodness of Fit of Model          Train Dataset
Classification Accuracy           : 100.00%
False Positive Rate               : 0.00%
False Negative Rate               : 0.00%

Goodness of Fit of Model          Test Dataset
Classification Accuracy           : 99.83%
False Positive Rate               : 0.09%
False Negative Rate               : 0.36%
```

Random Forest

```
Goodness of Fit of Model          Train Dataset
Classification Accuracy           : 99.93%
False Positive Rate               : 0.00%
False Negative Rate               : 0.25%

Goodness of Fit of Model          Test Dataset
Classification Accuracy           : 99.97%
False Positive Rate               : 0.00%
False Negative Rate               : 0.12%
```

Logistic Regression

```
Goodness of Fit of Model          Train Dataset
Classification Accuracy           : 99.43%
False Positive Rate               : 0.70%
False Negative Rate               : 0.25%

Goodness of Fit of Model          Test Dataset
Classification Accuracy           : 99.24%
False Positive Rate               : 1.01%
False Negative Rate               : 0.12%
```

Artificial Neural Network

Outcome

Optimal Threshold Probability

- False Postive Rate vs False Negative Rate
- Optimal threshold is: 79.234%

Categorising loan applicants

	Actual	Predicted
Low Risk	2172	2171
High Risk	839	840

Conclusion

Credit risk modeling is designed to enhance a bank's capability to forecast potential loan defaults among its customers. By examining past loan data and predicting future risks, the bank can implement measures to mitigate its overall risk and boost its profitability.