TCS Hackathon

Credit Default Predictor

Shubham Gupta
Somil Saxena
Saket Garodia
Vivek Sahoo

Contents

Problem at a glance

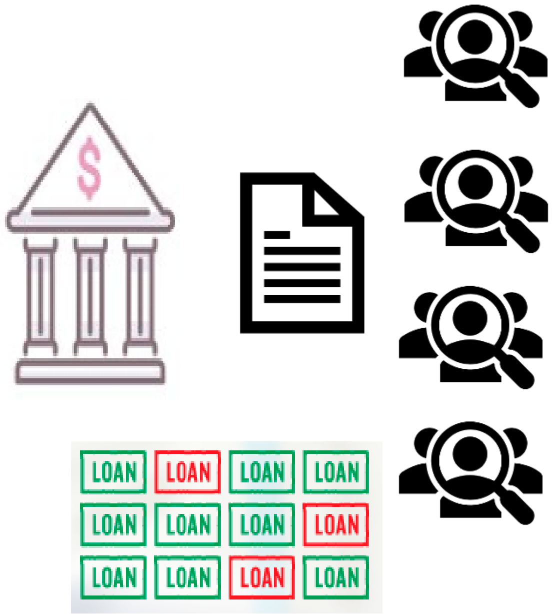
Data Source

kaggle

Data Set Size

150,000

Problem : **Improve on the state of the art in credit scoring by predicting the probability that somebody will experience financial distress in the next two years**



What bank has to do reduce default and improve its business??

Well, vast datasets.

Now, we have to come up with a model which can be useful for the banks

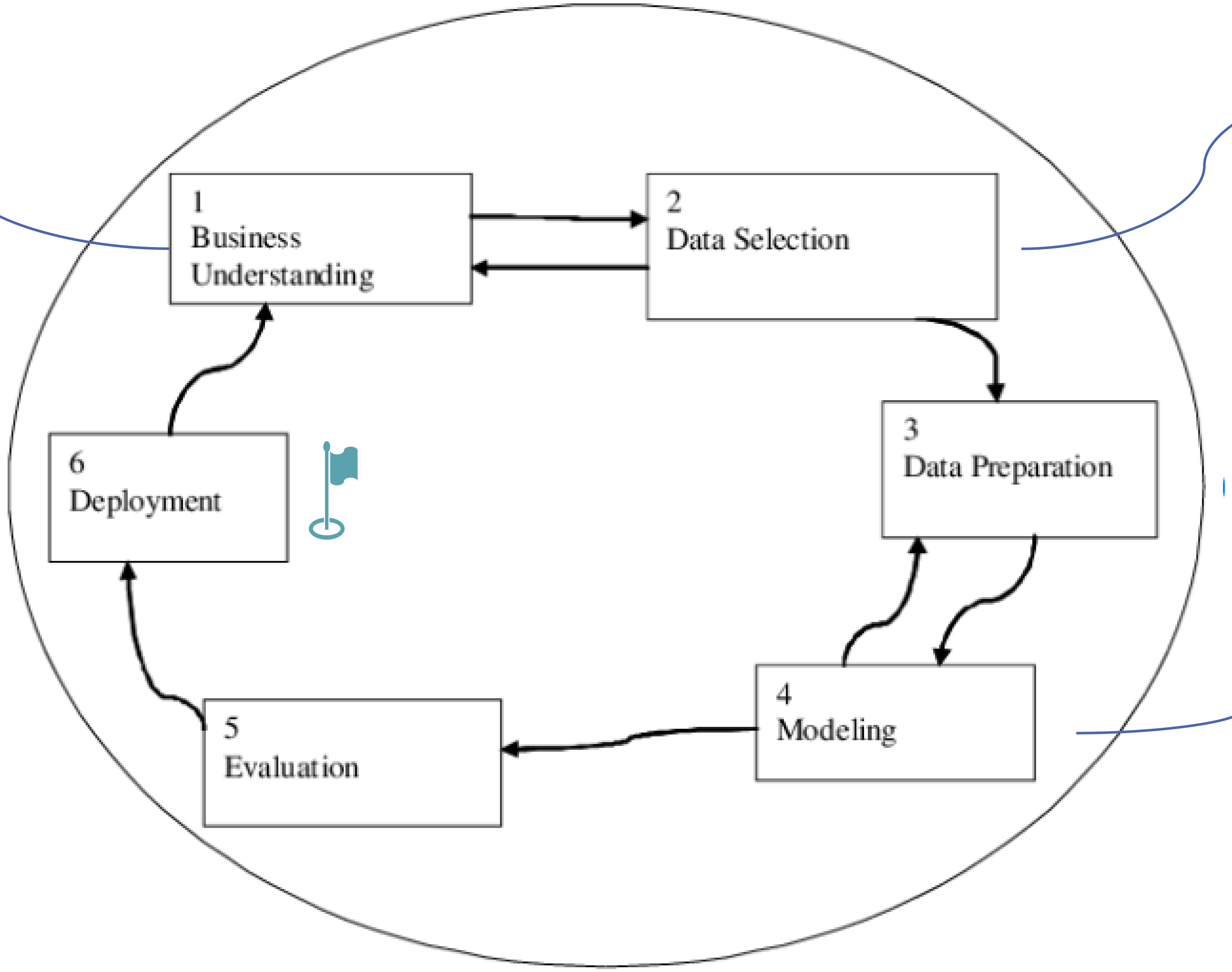
Data Mining
Black Box

4

Use Data Mining to learn customers leading to increase in Business

Data mining black box

- Understanding the consumer banking landscape
- Determining the business verticals that could leverage our Credit Risk Model



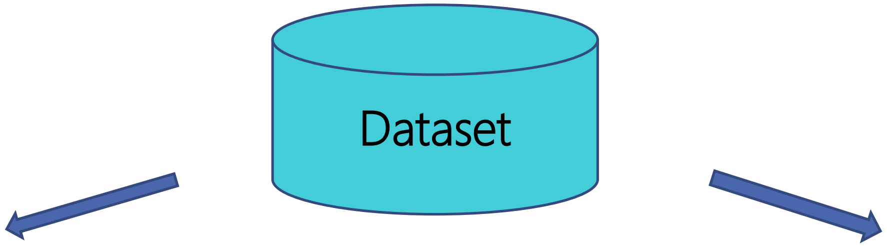
Understand, visualize and prepare it to feed into machine learning models

- Building a model to predict chances of delinquency in the next 2 years

Very important to look back at every step

4

Variables



Consumer spending behaviour

- Debt ratio
- Monthly Income
- Number of Real Estate Loans or Lines
- Revolving Utilization of Unsecured loans
- Number of open credit lines and loans
- Number of times (30-59 days) past due
- Number of times (60-89 days) past due
- Number of times 90 days past due

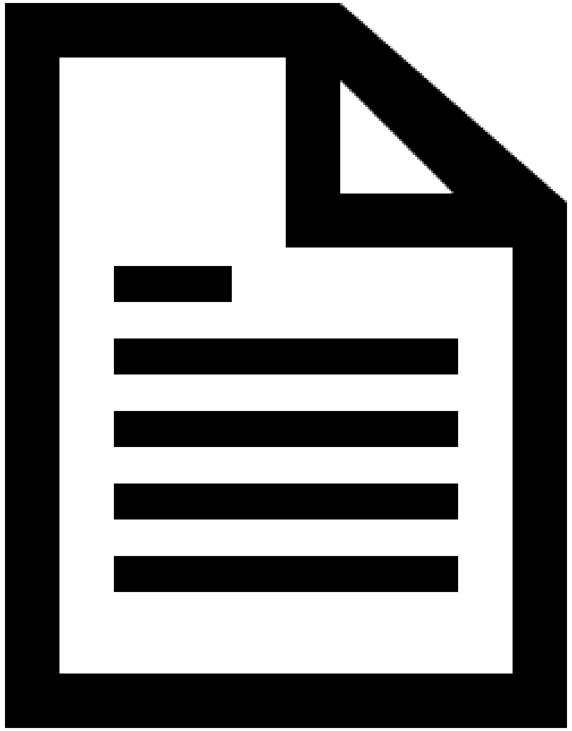
Consumer Characteristics

- Number of Dependents
- Age

Albert Einstein said,

I Would Spend 55 Minutes Defining the Problem and then Five Minutes Solving It

DATA PREPARATION



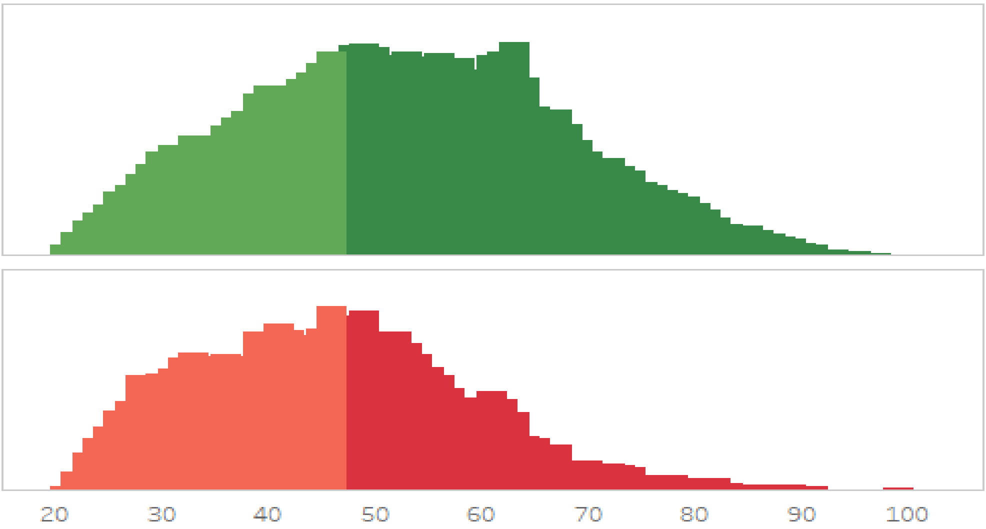
Exploring the variables

EDA

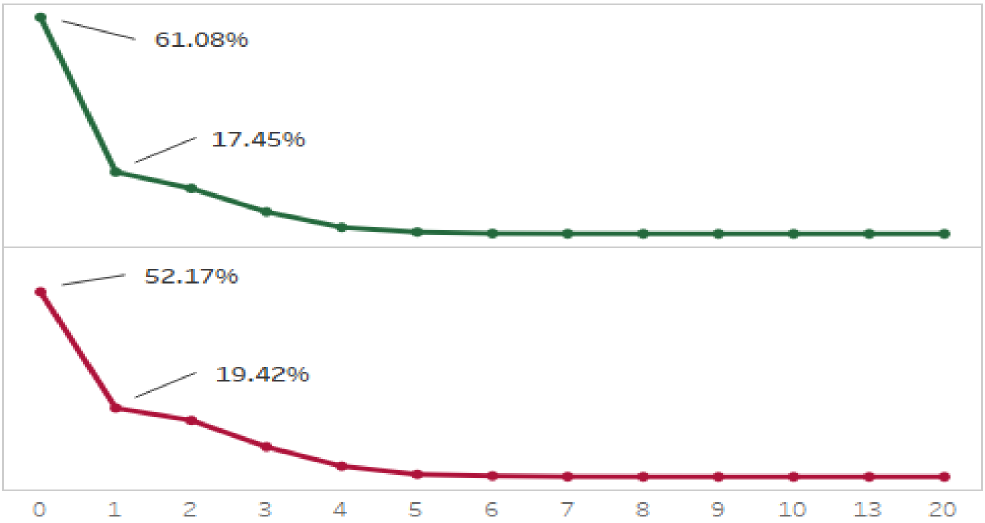
139,162
Serious Deliquency 0

9,800
Serious Deliquency 1

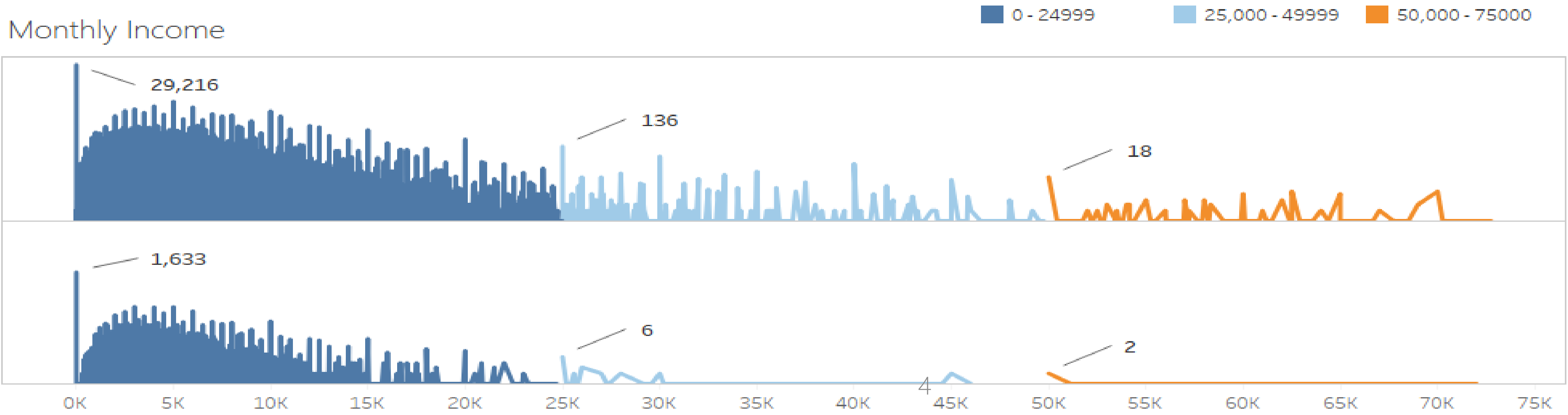
Age



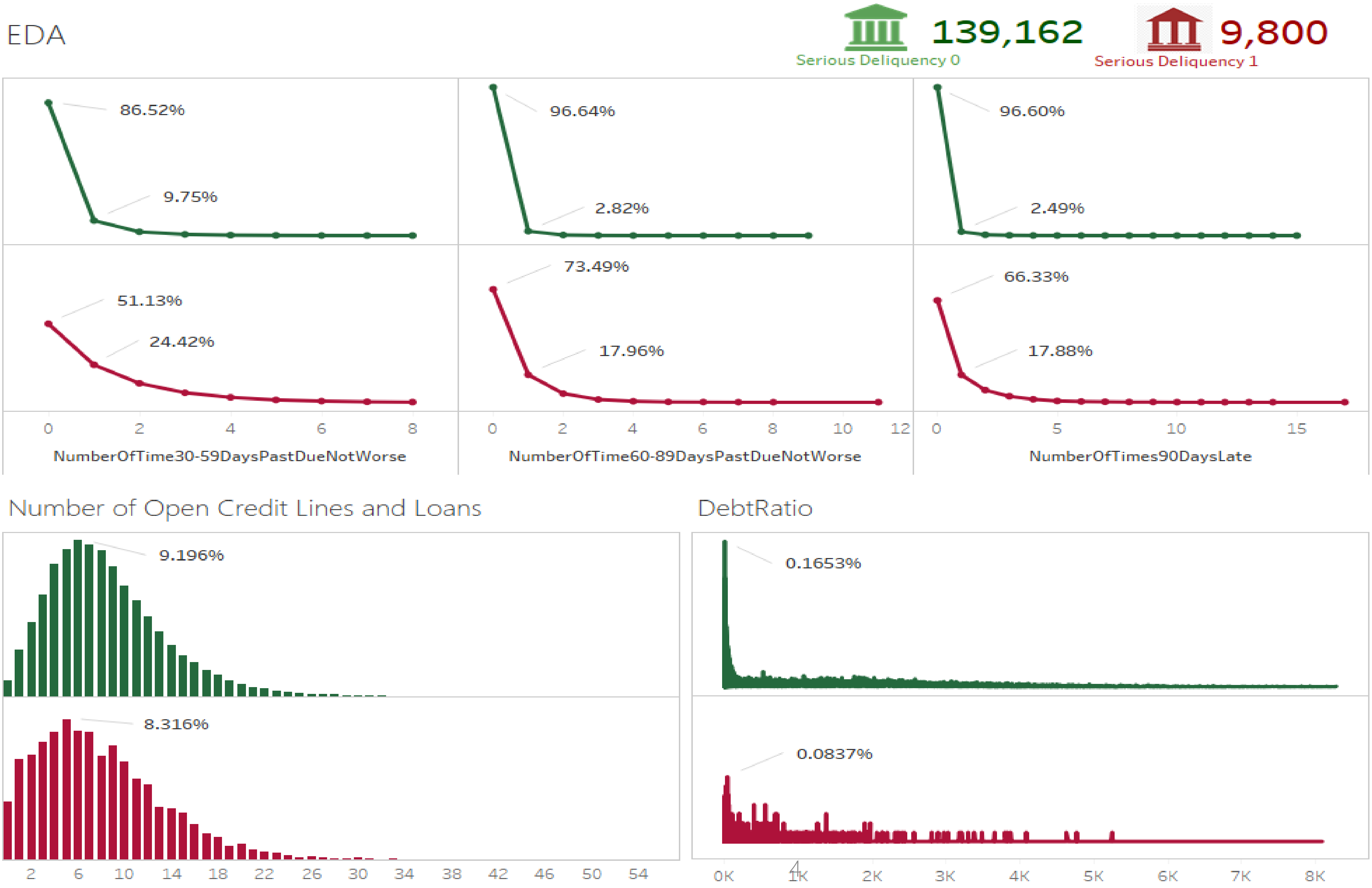
Number of Dependents



Monthly Income



Exploring the variables



DATA PREPARATION

Total customers
150,000

```
SeriousDlqin2yrs                        150000 non-null category
RevolvingUtilizationOfUnsecuredLines    150000 non-null float64
age                                     150000 non-null int64
NumberOfTime30-59DaysPastDueNotWorse    150000 non-null int64
DebtRatio                               150000 non-null float64
MonthlyIncome                           120269 non-null float64
NumberOfOpenCreditLinesAndLoans         150000 non-null int64
NumberOfTimes90DaysLate                 150000 non-null int64
NumberRealEstateLoansOrLines            150000 non-null int64
NumberOfTime60-89DaysPastDueNotWorse    150000 non-null int64
NumberOfDependents                      146076 non-null float64
```

Result after describing the data

Monthly income has 20% missing values

No of Dependents has 3% missing values

Next, we explored the distribution of all the other variables when the monthly income is null and not null

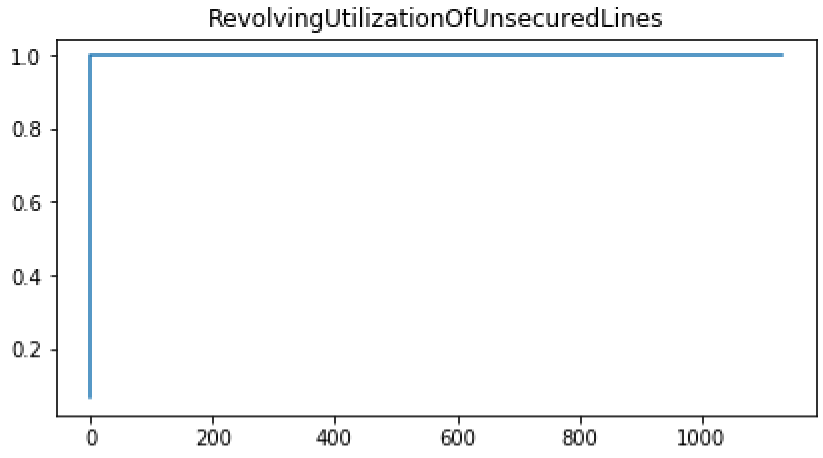
Monthly Income	Var1	Var2
Null		
Null		

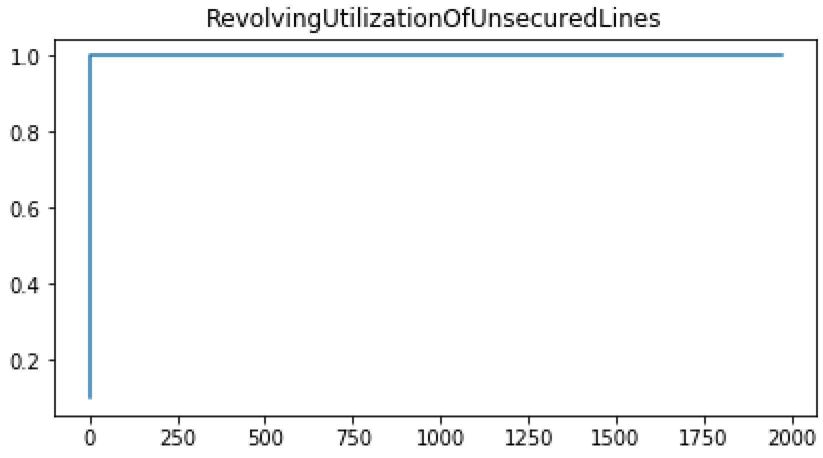
Monthly Income	Var1	Var2
Not Null		
Not Null		

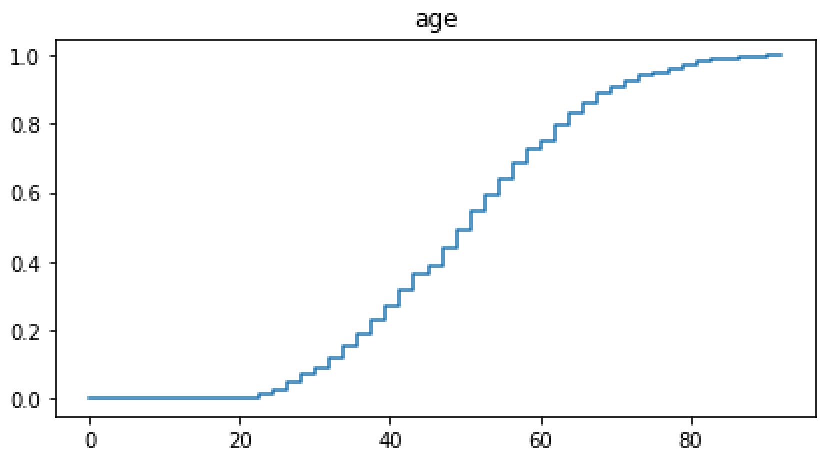
DATA PREPARATION

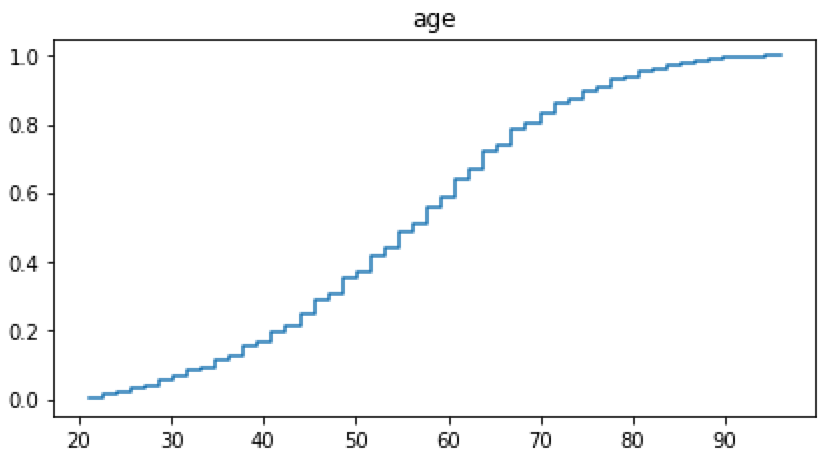
Monthly Income = not null

Monthly Income = null





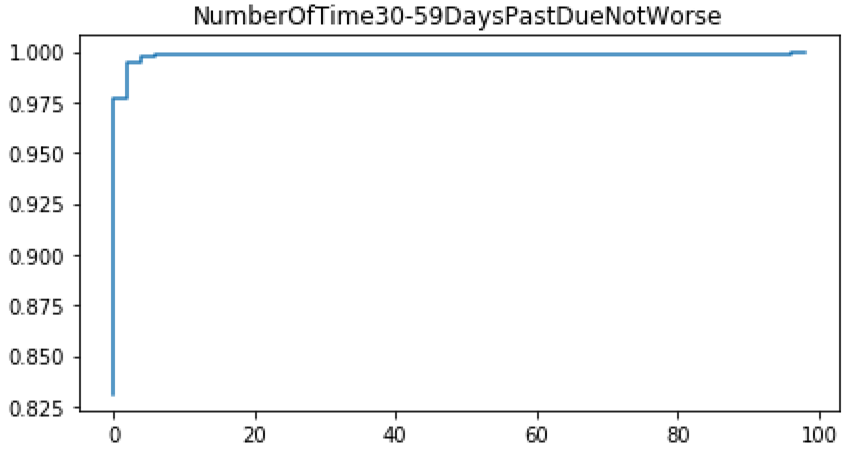


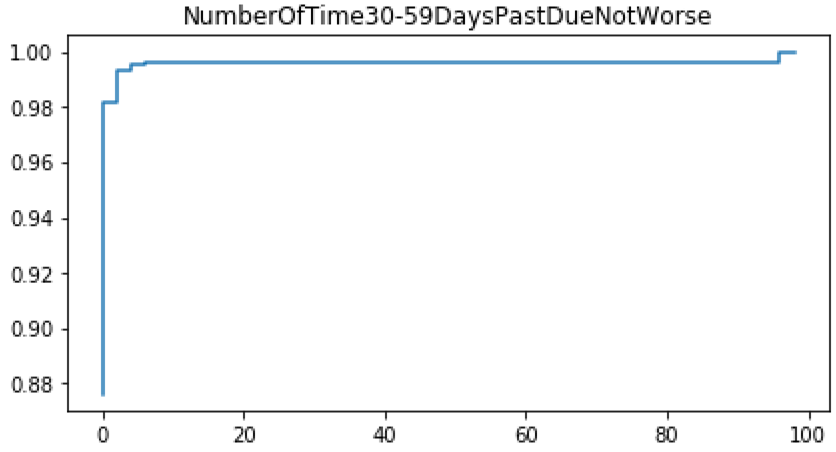


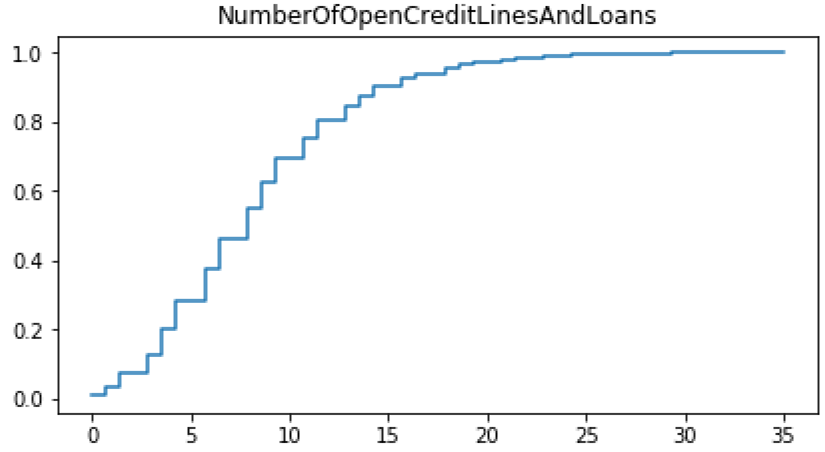
DATA PREPARATION

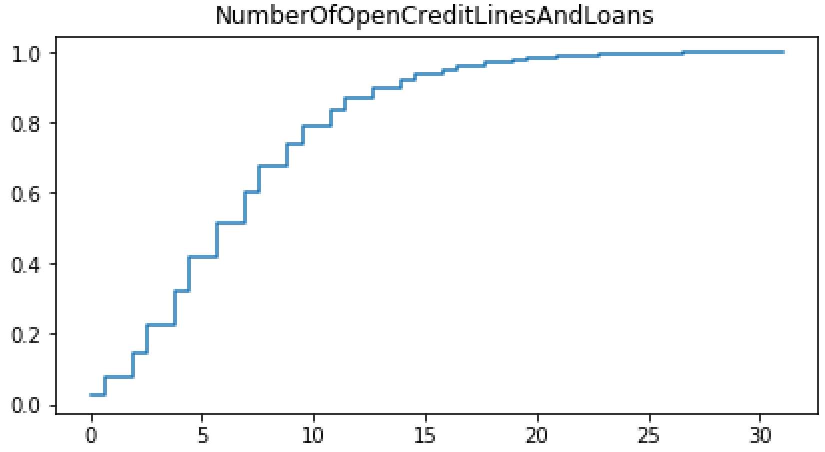
Monthly Income = not null

Monthly Income = null



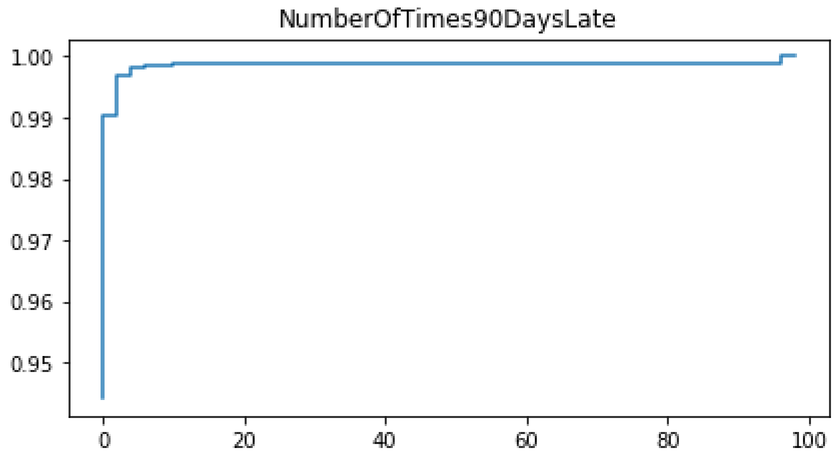


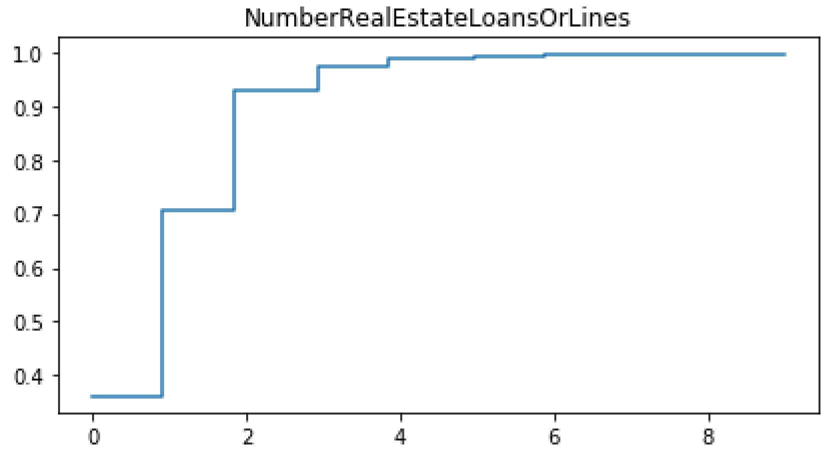




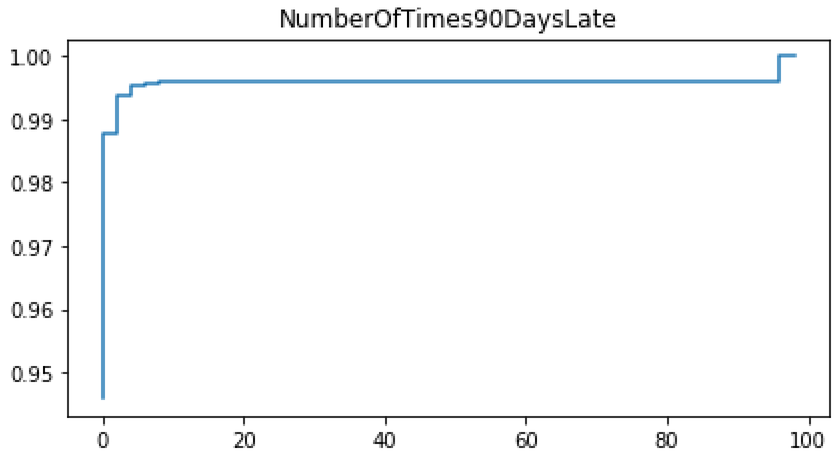
DATA PREPARATION

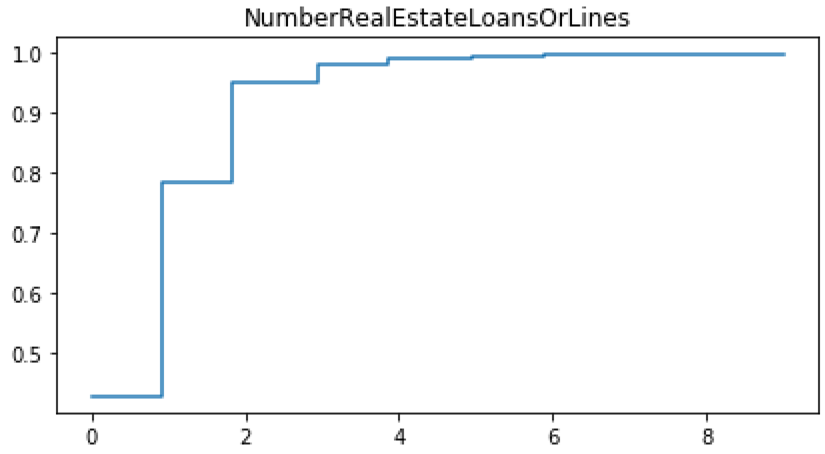
Monthly Income = not null





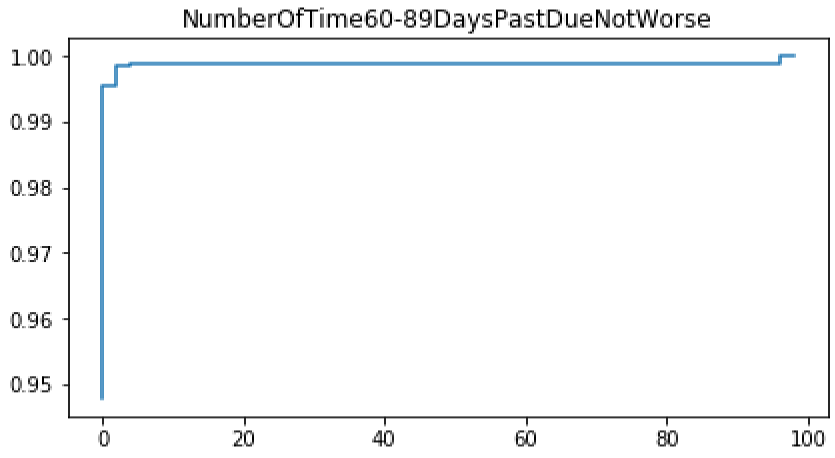
Monthly Income = null

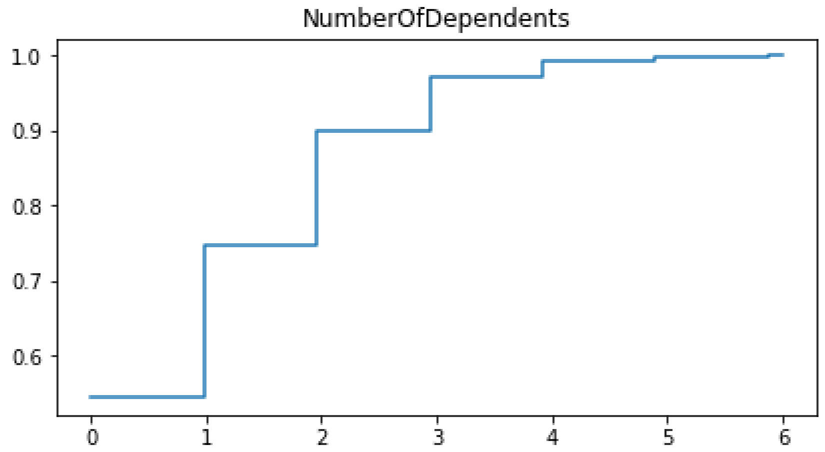




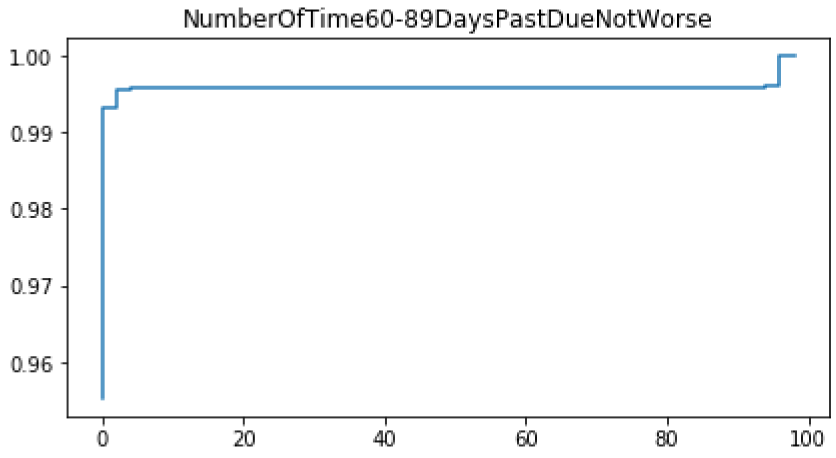
DATA PREPARATION

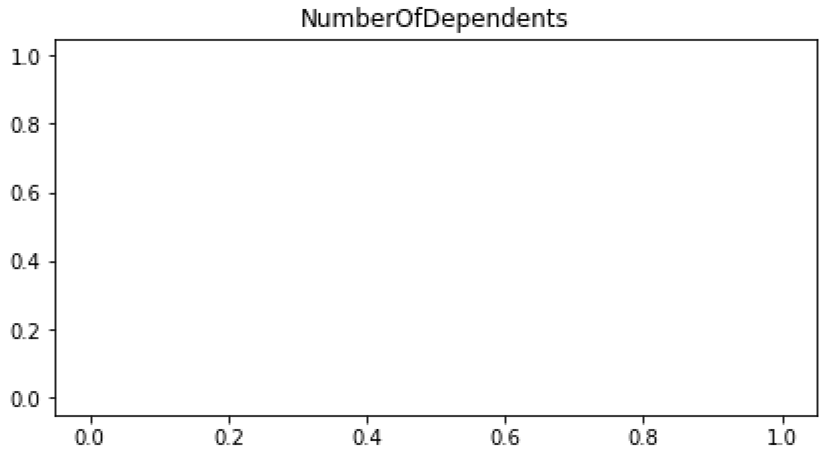
Monthly Income = not null





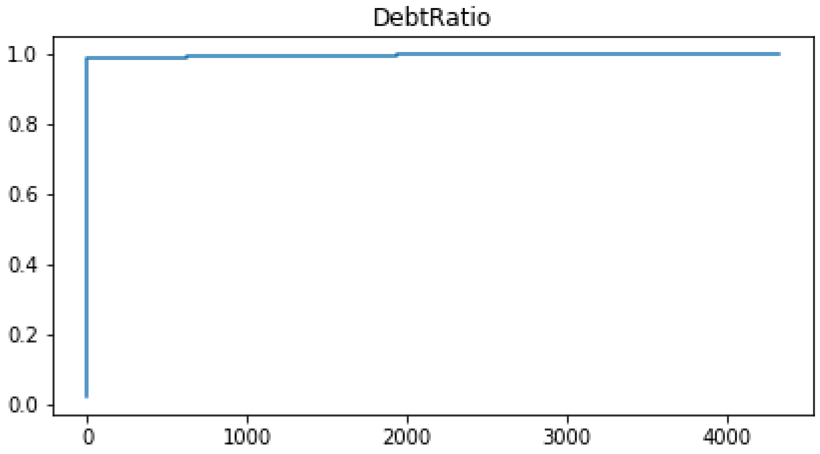
Monthly Income = null



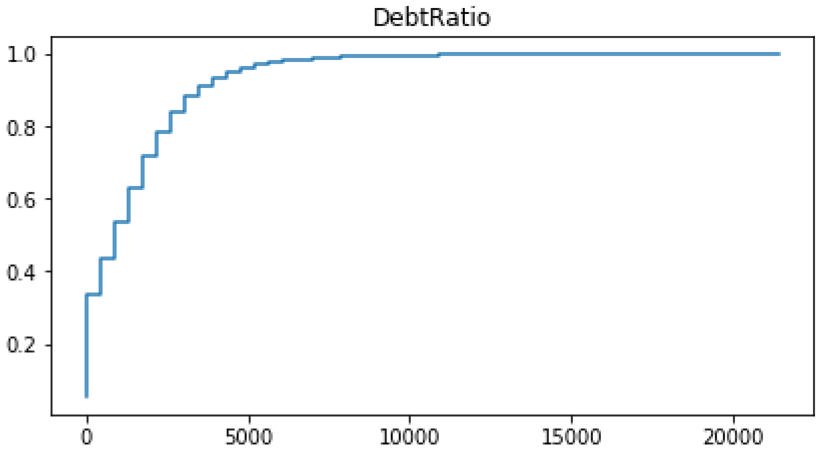


DATA PREPARATION

Monthly Income = not null



Monthly Income = null



Analysis from **distributions** ?

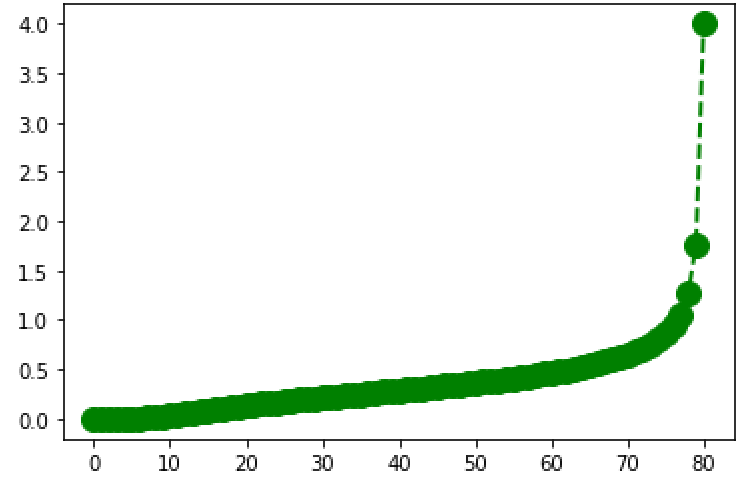
The Monthly income values which have '*null*' values have different corresponding **DebtRatio** distribution compared to '*not null*' values.

<u>Result</u> : Debt Ratio distribution can help in imputing null values of monthly income

DATA PREPARATION

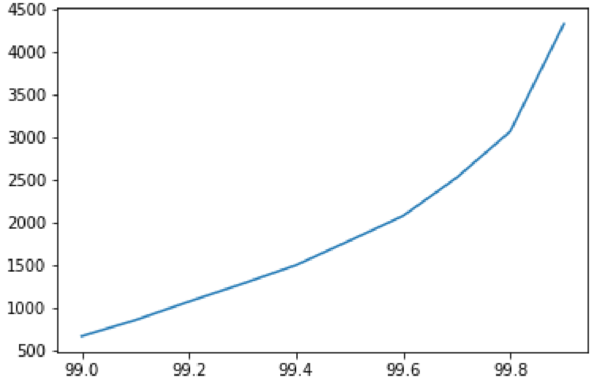
Distribution of Debt_ratio

[<matplotlib.lines.Line2D at 0x7f789d467710>]

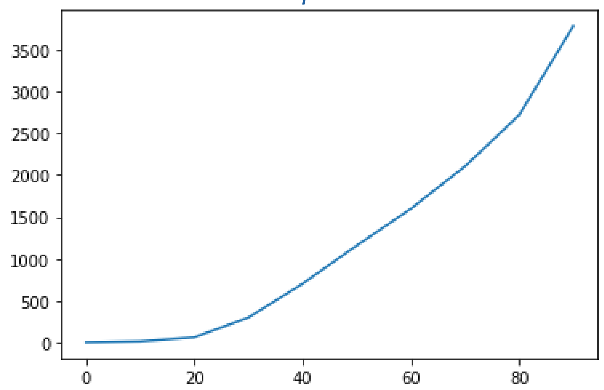


Deeper analysis

99-100 percentile



0-100 percentile



Result : Missing income will have the same distribution as the top 1% debt ratio, so we will impute the missing income with the missing income for the top 1% debt ratio

We got 0.2391667, so imputing with 0. Its intuitive too,
high debt_ratio implies *Less Income*

Note : For number of dependents, there's the same distribution of the variables when they have null and non-null values, so imputing with the *median* which is 0

DATA PREPARATION

Outliers analysis

Boxplot outlier analysis
Values > 1.5 IQR

```
NumberOfOpenCreditLinesAndLoans           3980
age                                         46
NumberOfTimes90DaysLate                   8338
DebtRatio                                31311
NumberOfTime60-89DaysPastDueNotWorse      7604
NumberRealEstateLoansOrLines               793
NumberOfTime30-59DaysPastDueNotWorse     23982
MonthlyIncome                                0
RevolvingUtilizationOfUnsecuredLines       763
NumberOfDependents                           0
dtype: int64
```

Almost 40% of the data have some outliers, so we cannot remove them all, lets delve deeper

0	126018
1	16033
2	4598
3	1754
4	747
5	342
6	140
7	54
8	25
9	12
10	4
11	1
12	2
13	1
96	5
98	264

NumberOfTime30-59DaysPastDueNotWorse

0	141662
1	5243
2	1555
3	667
4	291
5	131
6	80
7	38
8	21
9	19
10	8
11	5
12	2
13	4
14	2
15	2
17	1
96	5
98	264

NumberOfTimes90DaysLate

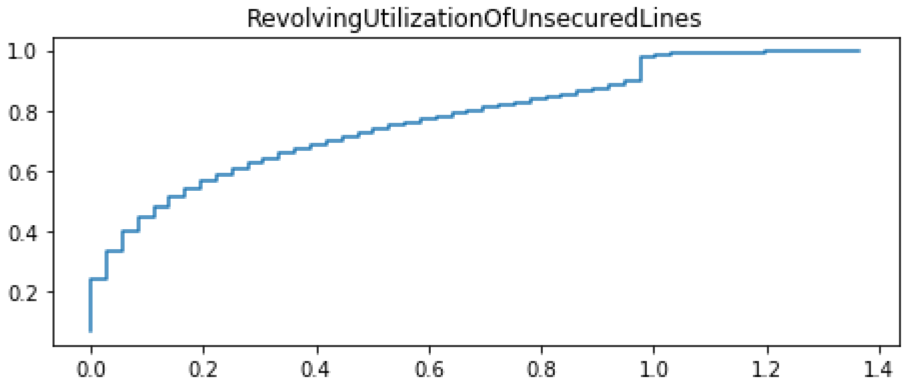
0	142396
1	5731
2	1118
3	318
4	105
5	34
6	16
7	9
8	2
9	1
11	1
96	5
98	264

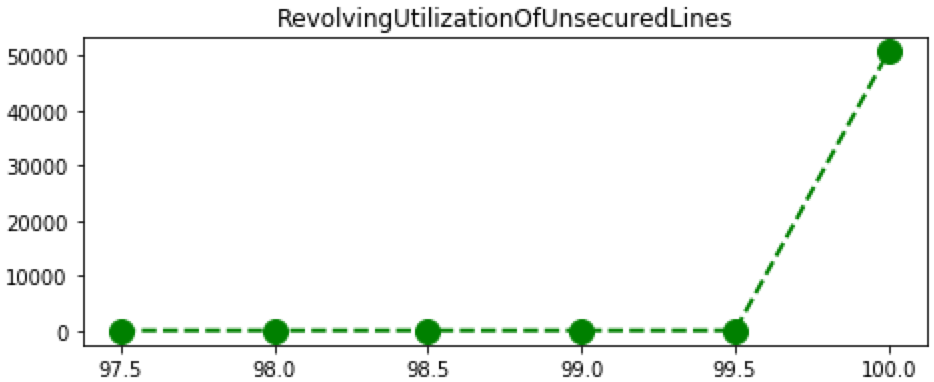
NumberOfTime60-89DaysPastDueNotWorse

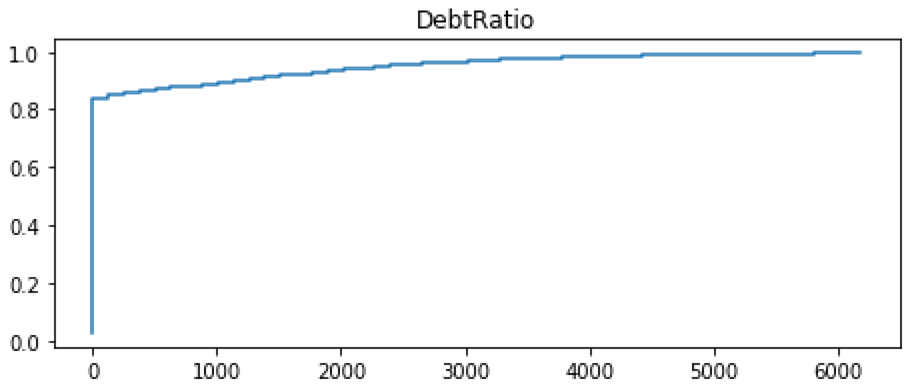
Result : All these 3 variables have most of the value below 20 except 96 and 98 that occur 5 and 264 times in all the 3 datasets.
There might be some error or maybe an actual data but anyways a outlier
We decided to impute those values with 20

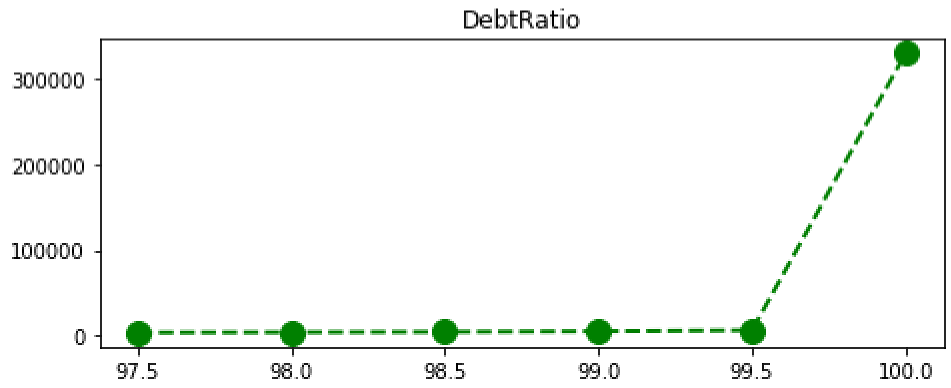
In a similar fashion, we capped *NumberRealEstateLoansOrLines* to 30 and *NumberOfOpenCreditLinesAndLoans* to 40

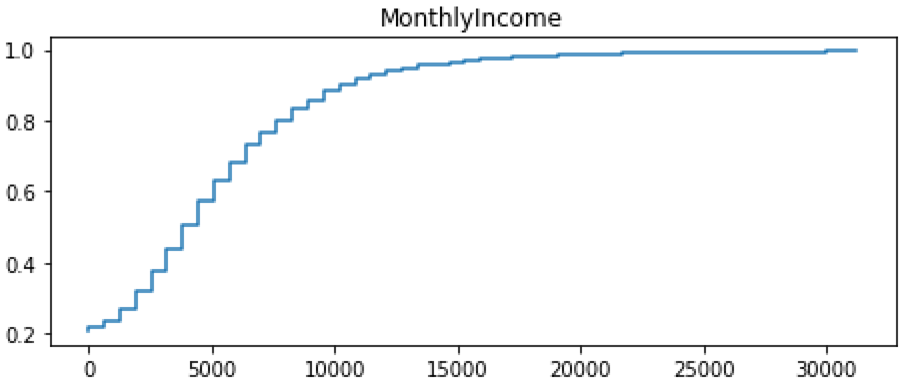
DATA PREPARATION

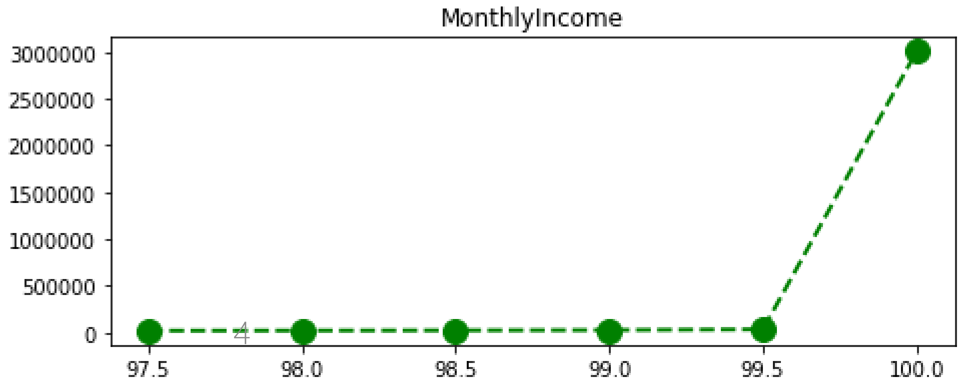












Result:
For these 3 variables, after looking at the ECDF and percentile plots its clearly evident that the values shoot up after the 99.5th percentile

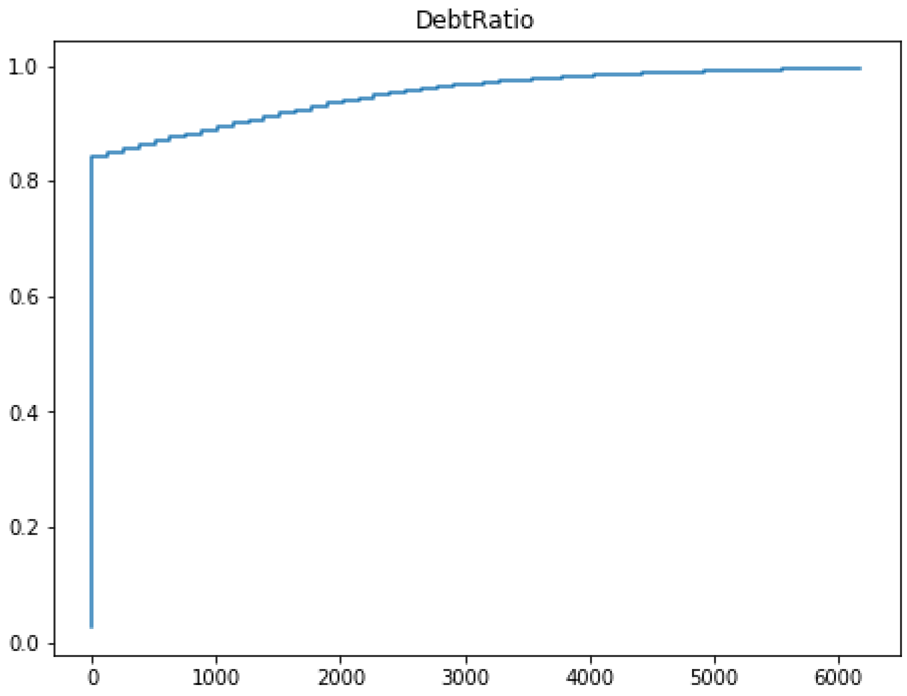
Solution: *We removed top .5% values for all of these.*

In 150,000, .5% wouldn't matter much and outlier removals and capping is a must to generalize the datasets else it might overfit

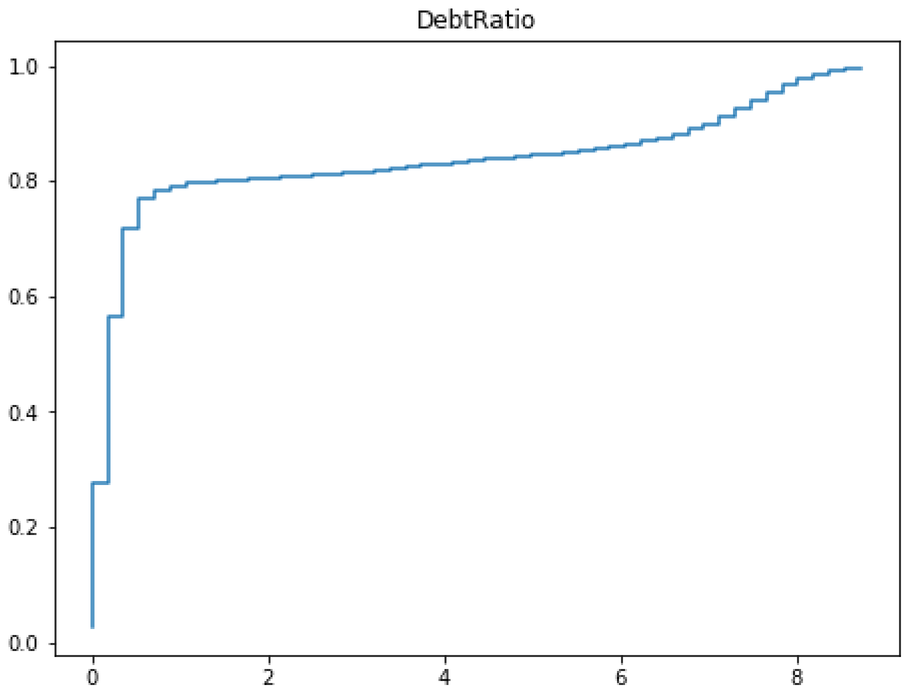
DATA PREPARATION

On our final step, we just smoothened our debt ratio column so that the machine learning models handle it well, since the ECDF was jumping after some low values.

Before smoothing



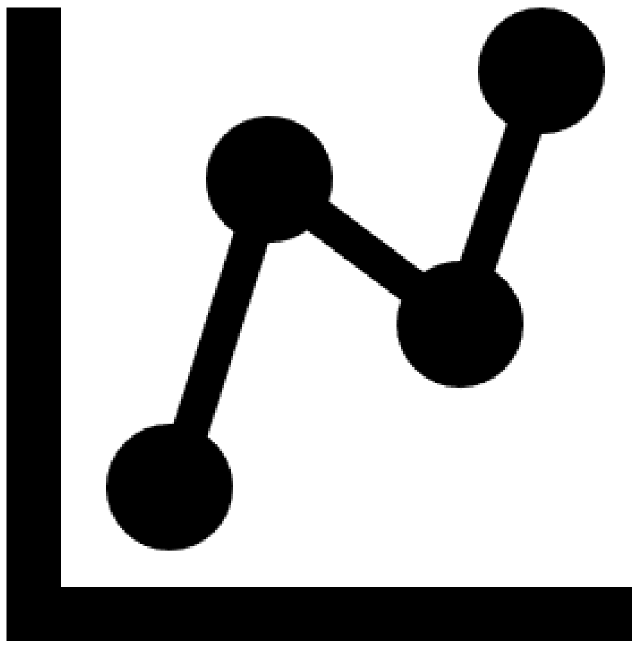
After smoothing



"As data piles up, we have ourselves a genuine gold rush. But data isn't the gold. I repeat, data in its raw form is boring crud. The gold is what's discovered therein."

— Eric Siegel, Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die

PREDICTIVE MODELLING



PREDICTIVE MODELLING

Clean data

It's a supervised classification problem
Challenge?? Unbalanced dataset

Only **7%** of our training data has Serious Delinquency as 1

Intuition before model building : We cannot chose models based on accuracy
Why? Even if we build a dummy model that predicts 0 for any new dataset it will have a accuracy of **93%** because the chance of delinquency is only **7%**

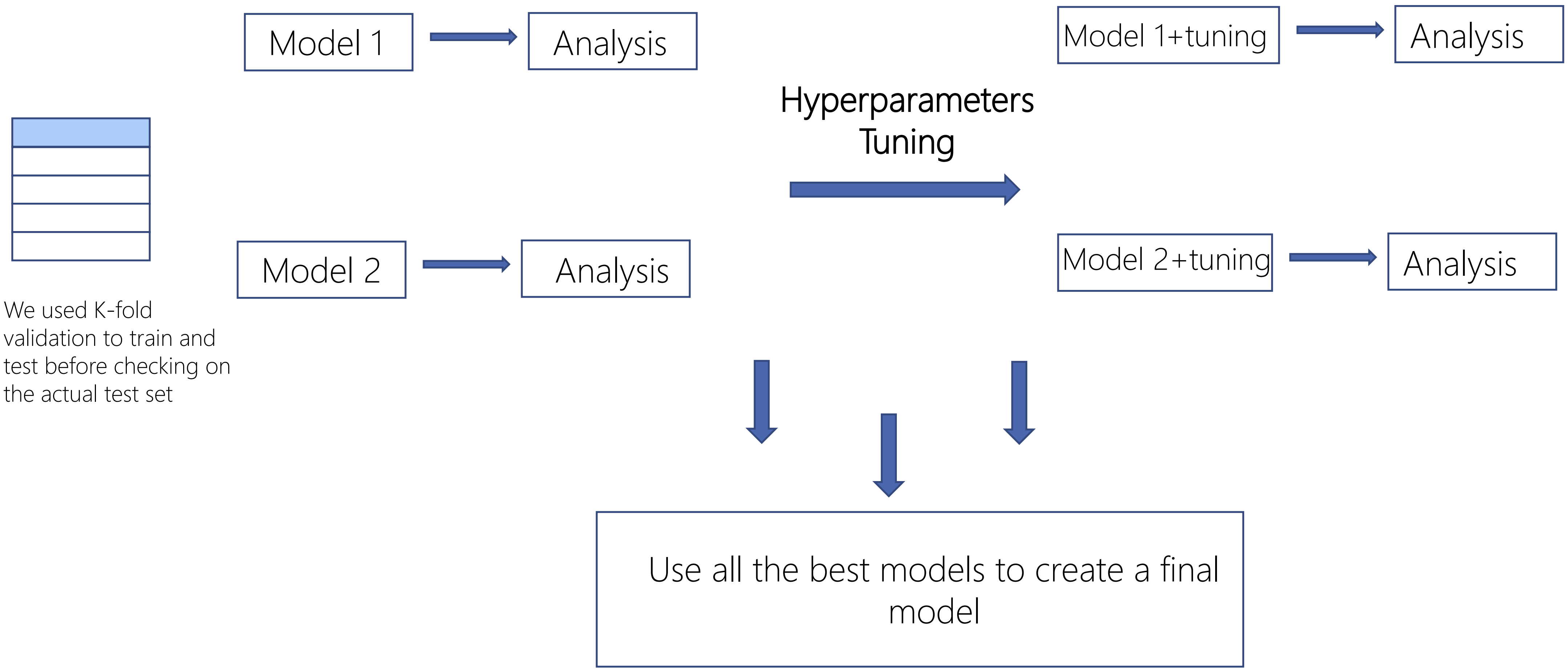
Confusion matrix for dummy model

	Predicted -	Predicted +
Actual -	34749	0
Actual +	2492	0

	precision	recall
0	0.93	1.00
1	0.00	0.00

Conclusion: We will look for other metrics like AUC, Precision, Recall to build the best model

PREDICTIVE MODELLING



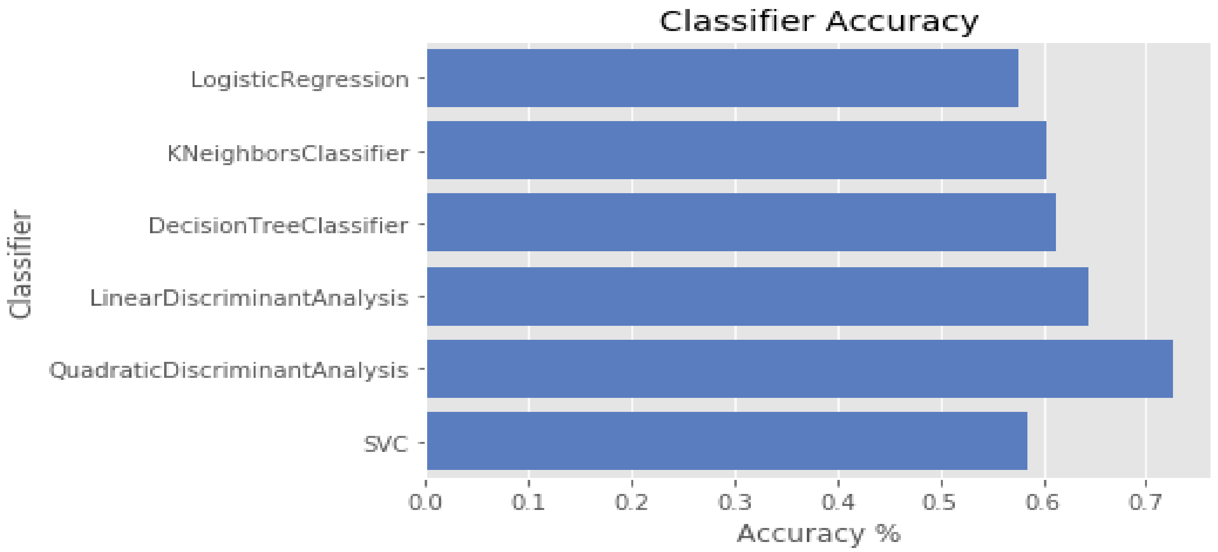
PREDICTIVE MODELLING

Step 1: 11 predictor variables → Tested if we can compress into a few variables using PCA

Result : PCA couldn't explain the variance just using 2-3 principal components
So, taking all variables

Step 2 : Built different available classifier models using k-fold validation

- ❑ Logistic Regression
- ❑ K- Neighbors Classifier
- ❑ DecisionTreeClassifier
- ❑ SVC
- ❑ Linear Discriminant Analysis
- ❑ Quadratic Discriminant Analysis



Result : These models independently didn't perform very good based on AUC score

PREDICTIVE MODELLING

Step 3: Used one of the most powerful machine learning libraries – Ensemble models

- ❑ Random Forest
- ❑ Gradient Boosting gave us an **amazing AUC of .85826 on Kaggle's test set**

Your most recent submission

kaggle

Name	Submitted	Wait time	Execution time	Score
submission_check.csv	6 days ago	0 seconds	1 seconds	0.85826

Complete

In the money   Gold   Silver   Bronze

#	∆pub	Team Name	Notebook	Team Members	Score	Entries	Last
1	▲1	Perfect Storm			0.86955	128	8y
2	▲4	Gxav			0.86929	54	8y
3	▲14	occupy			0.86928	9	8y
4	▲16	D'yakonov Alexander (MSU, M...			0.86919	64	8y

Why ensemble is always better when there is no infrastructure constraint??

Because it reduces bias as well as variance by taking the power of many models and improving the errors of previous ones

PREDICTIVE MODELLING

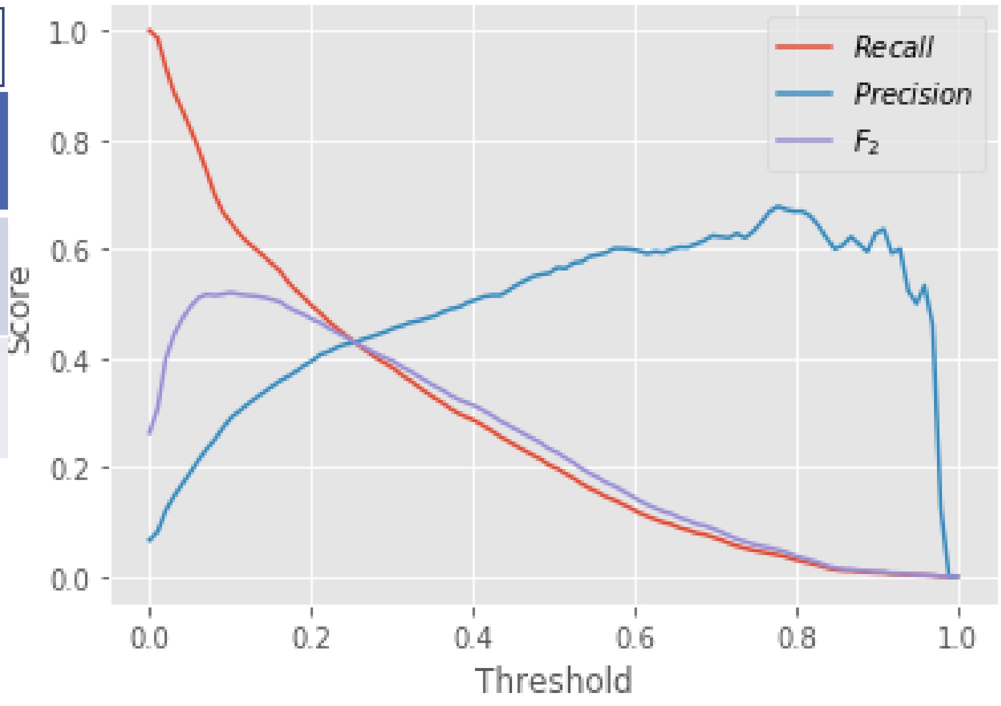
Deeper Analysis of our best model – Gradient Boosting

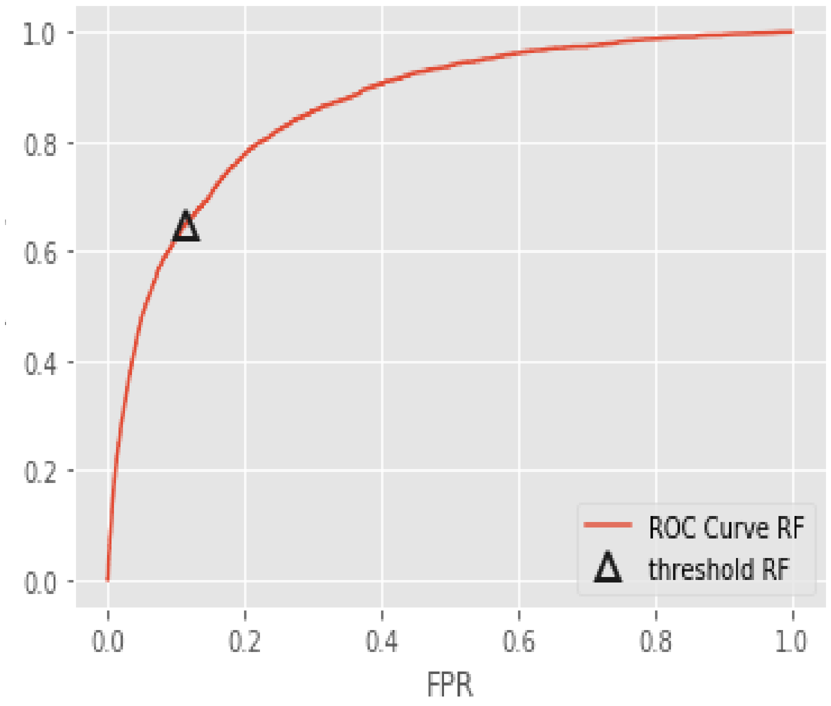
Confusion Matrix

	Predicted negative	Predicted positive
True negative	41527	470
True positive	2399	604

Precision = $\frac{True\ Positive}{True\ Positive + False\ Positive}$

Recall = $\frac{True\ Positive}{True\ Positive + False\ Negative}$





- ❑ Precision = .56
- ❑ Recall = .20

20% of the people in the test set who actually committed a serious delinquency were classified as people who could commit

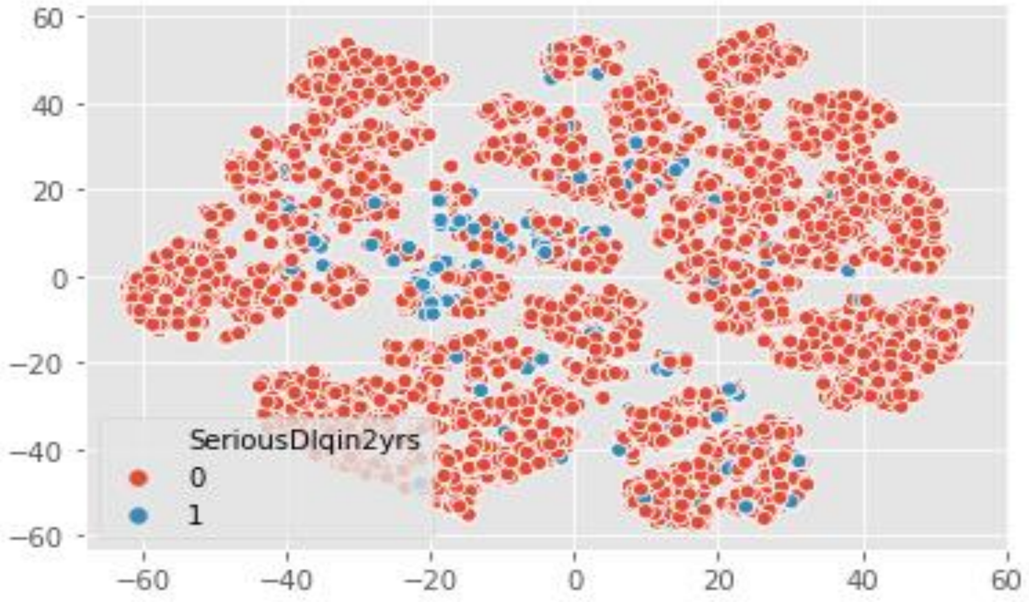
56% of the people in the test set for whom the model predicted as highly probable for a chance of commiting serious delinquency did actually commit

How can the model be useful to business?

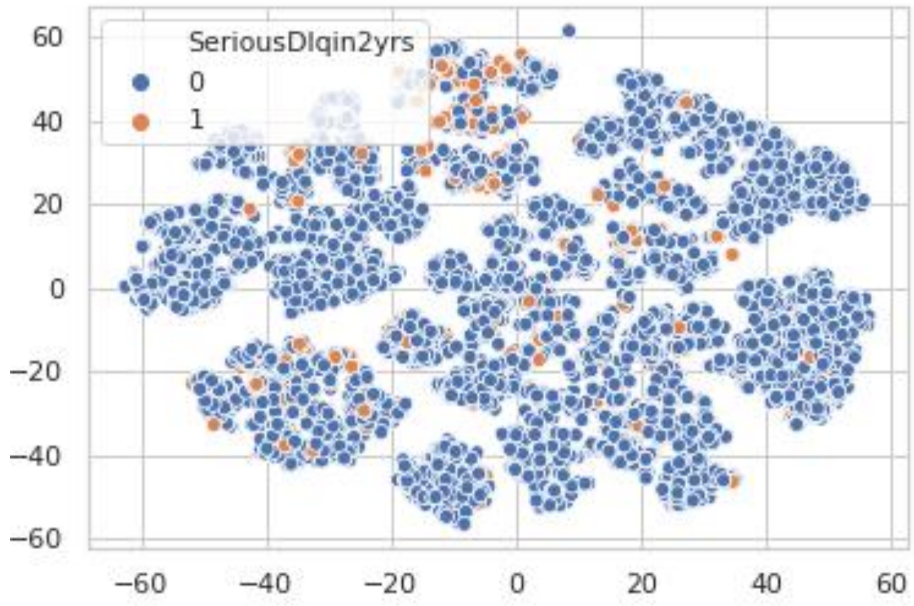
4

PREDICTIVE MODELLING

Distribution on 2-dim using 11 features



We can get an intuition that the distribution of the datasets of 1s and 0s are so closely distributed.



Now, we tried a **deep learning technique Autoencoder** to convert the 11 features to 50 features

Result : The distribution is still very close

Interpretation of the resulting metric

High risk appetite

Cares less about recall, so focus on precision. Why? Low appetite for default
Increase the probability threshold to flag as a defaulter

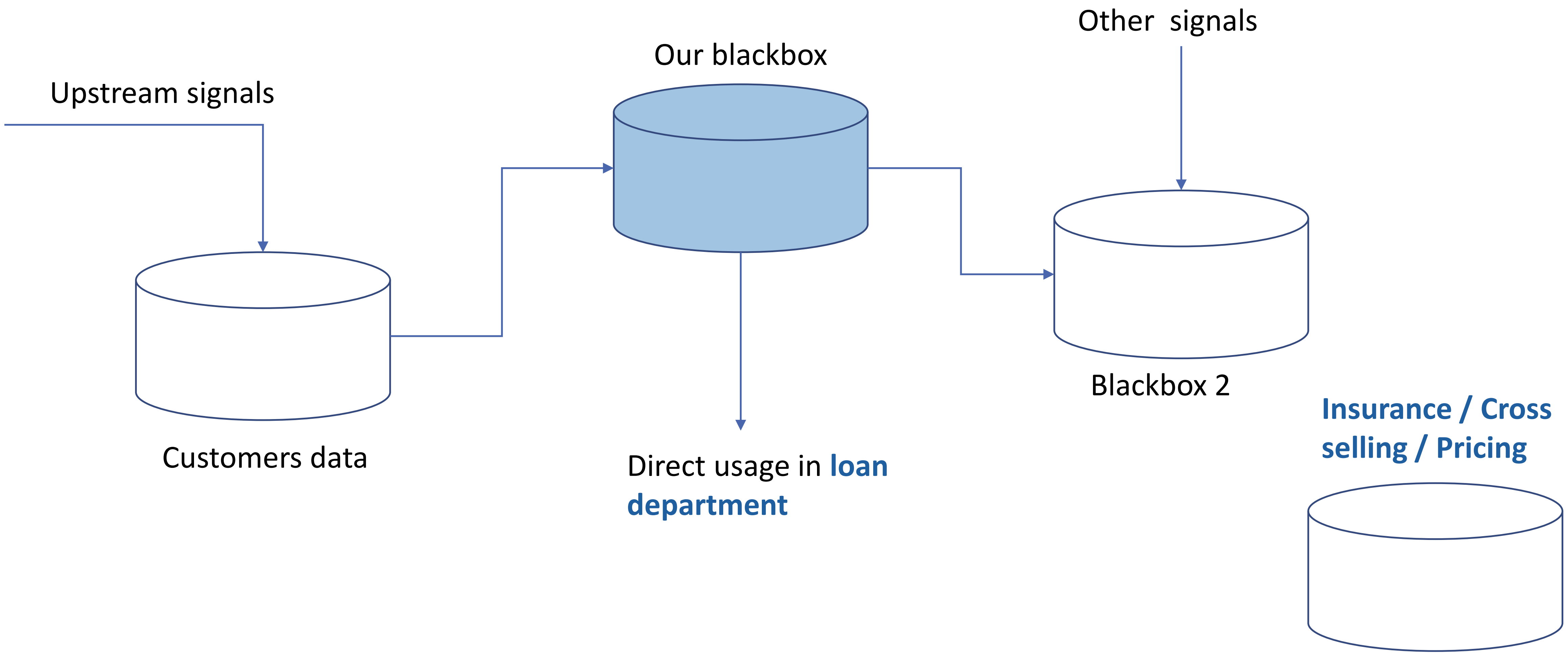
Low risk appetite

Needs more recall value. Why? Low appetite for default
Decrease the probability threshold to flag as a defaulter

Precision vs Recall tradeoff value

Using model in other pipelines

One machine learning pipeline (or Black box) may not directly be used to affect decision making



Thank you

Questions ?