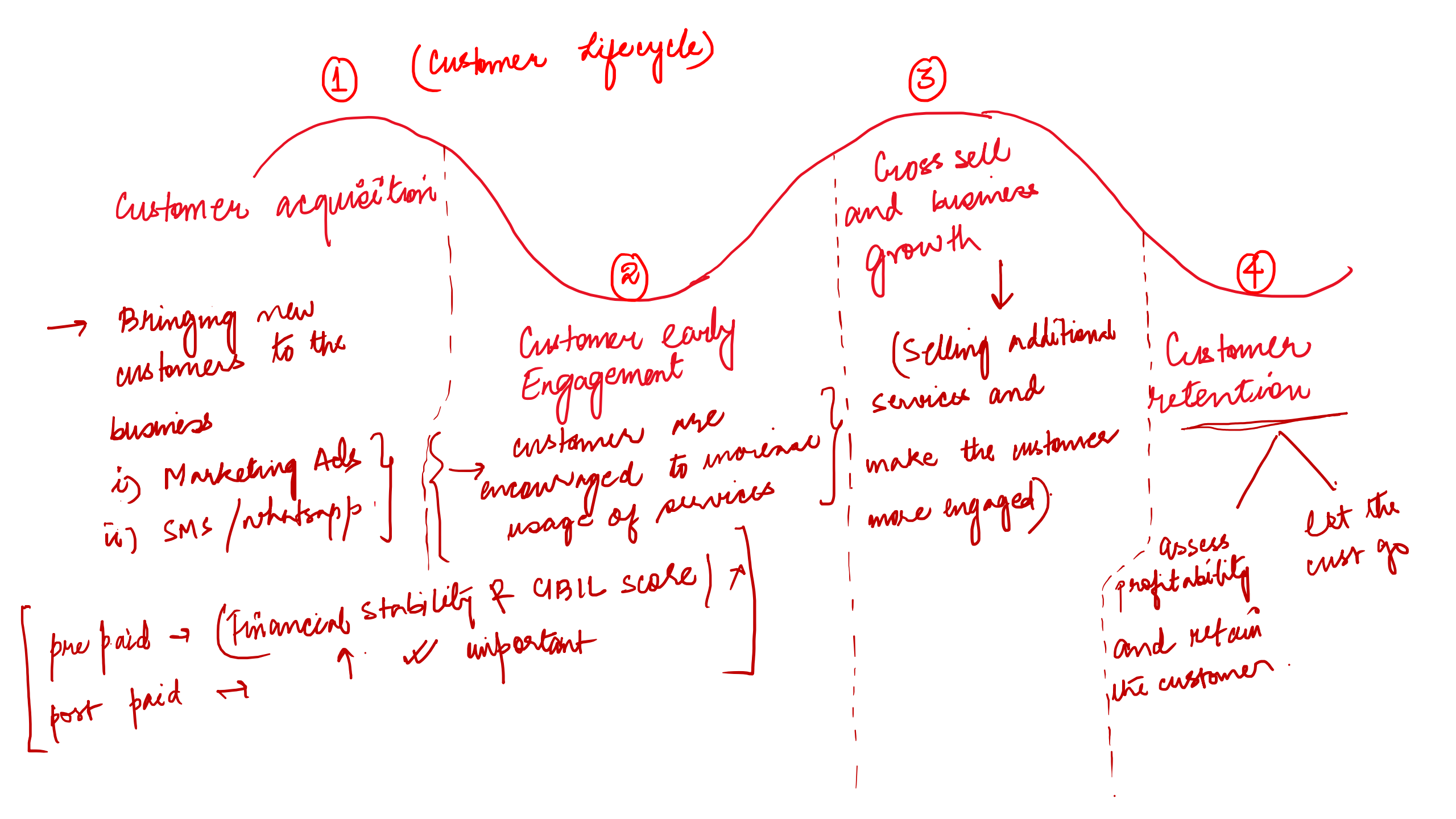
Project Topic → "Telecom Churn Modeling"

8-10 AM

Approach → ① Theory and approach
② Python Implementation


(Customer lifecycle)
1
Customer acquisition
→ Bringing new customers to the business
i) Marketing Ads
ii) SMS / whatsapp
2
Customer early Engagement
→ customer are encouraged to increase usage of services
[pre paid → (Financial stability & CIBIL score) ↑ important
post paid →]
3
Cross sell and business growth
↓
(Selling additional services and make the customer more engaged)
4
Customer retention
assess profitability and retain the customer
Let the cust go

Customer Acquisition

i) Come up with features or factors that can be predictive of Customer value

eg. CIBIL score

Higher CIBIL score, means better past paid customer

Q. Who get the past paid offer?

Early Eng

ii) What can be done to increase usage of services?

Q. Which set of treatments can improve engagements?

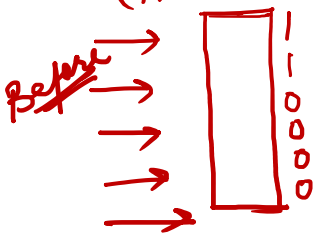
→ digitally → Targeted Ads.

Cross sell

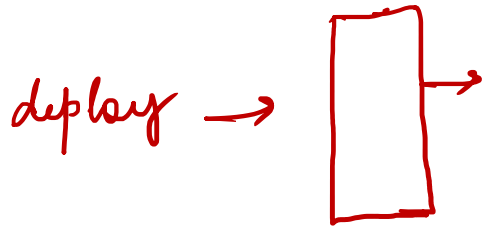
iii) What's additional that can be sold?

① Build a response model from earlier campaigns

(Past campaign) Dec'20



Present day (May'21)



Customer Retention

Attrition → (Canceling any ongoing relationship)

Reactive

i) Customer calls in and wants to cancel

↙

(A) Profitability assessment

(B) Offers will be made

"Pro active"

i) Predicting which customers are likely to attrite

↓

(Our use case for the project)

(Project Objective) → Develop a predictive framework for enabling proactive retention strategy for a Telecom company

Project Presentation flow :- (For interviews)

1. speak about the impact first (Potential Benefit)
   mention what was enabled by your solution. for e.g. 30% reduction in cost. OR 20% increase in profit.
2. which part of lifecycle is addressed by your solution?
3. Justify the approach (supervised / unsupervised)
4. steps performed. to solve the problem.

Motivation → " Reduce the churn rate and improve retention "

Outlier analysis
Bivariate analysis

Missing Value Imputation for Numerical Features

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
num_1=pd.DataFrame(imputer.fit_transform(num),index=num.index,columns=num.columns)
```

Missing Value Imputation for Categorical Features

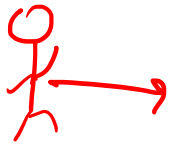
```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='most frequent')
char_1=pd.DataFrame(imputer.fit_transform(char),index=char.index,columns=char.columns
```

Some useful concept to be used in Python Implementation

Databases:

i) Automated capture of data.

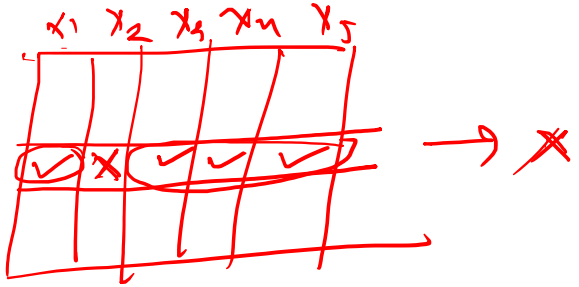
ii) Tools are used to scan forms, OCR etc

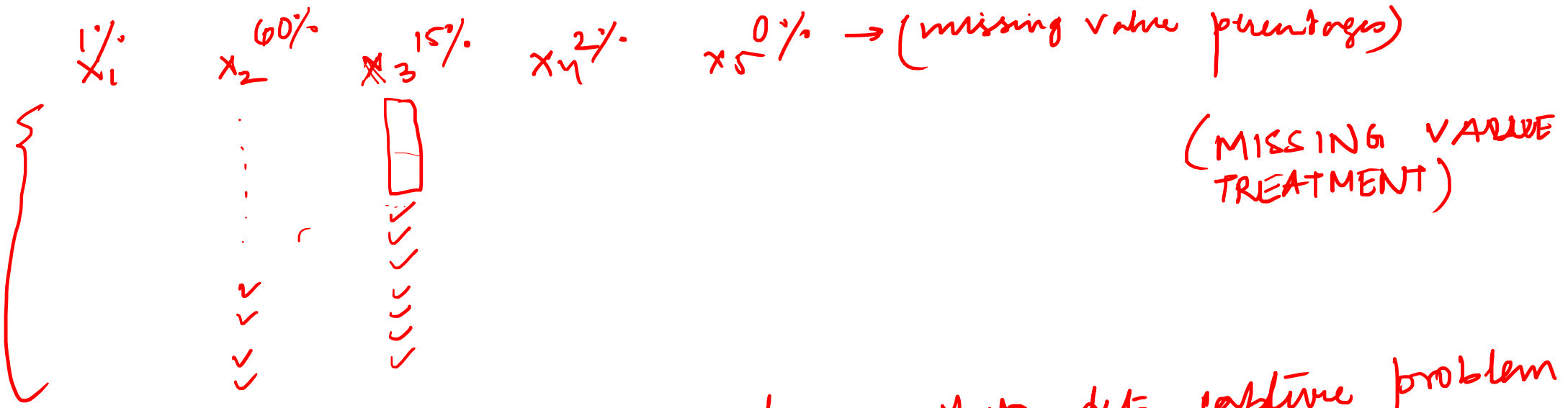


→ CC → pay reported / not reported → missing value

optional.

→ The business records a non activity as missing value. (missing value carries a meaning)



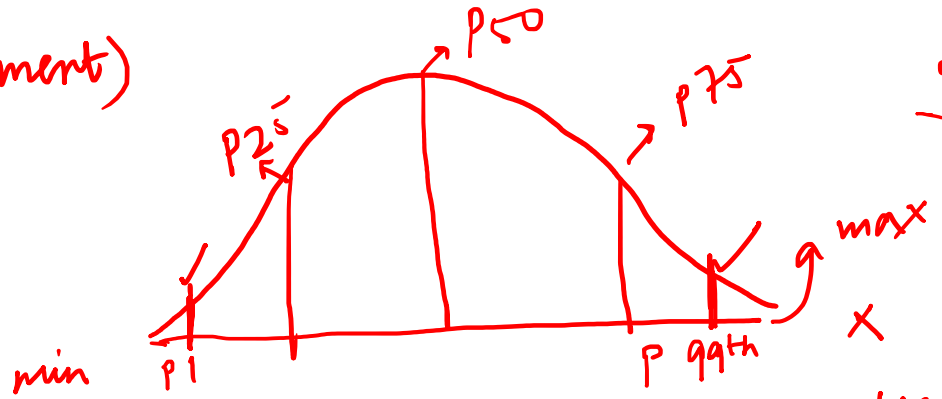


(A) As a data scientist I do not know what data capture problem led to missing values.

(B) If a feature records more $\frac{1}{4}$ of its rows as missing, then it is best to use such a feature as a <u>last resort</u>

(C) $E(\bar{X}) = \mu$

(Outlier treatment)



Distribution of feature x

→ (As variance of a feature reduces the predictive power also reduces) **

→ Treat the extreme values in such a way that variance reduction is kept at a minimum

If $x < p1$ then $x = p1$ else x } → FLOORING

If $x > p99$ then $x = p99$ else x } → CAPPING

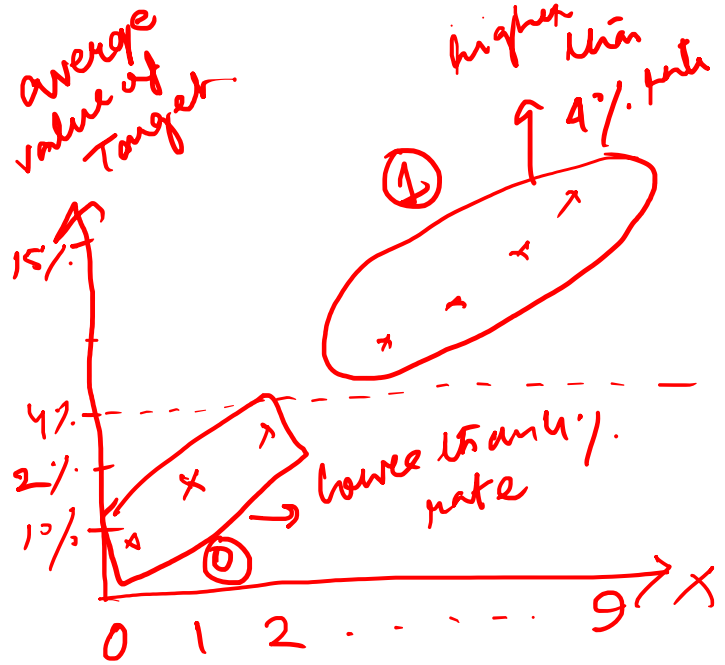
Bivariate Analysis

White Box method for selecting features.

Age	Age Ranks (0–9)	Outcome (Target)
a_1	0	1
a_2	0	0
a_3	1	1
⋮	2	0
	⋮	⋮
a_n	9	0

mean value = 4% (suppose)

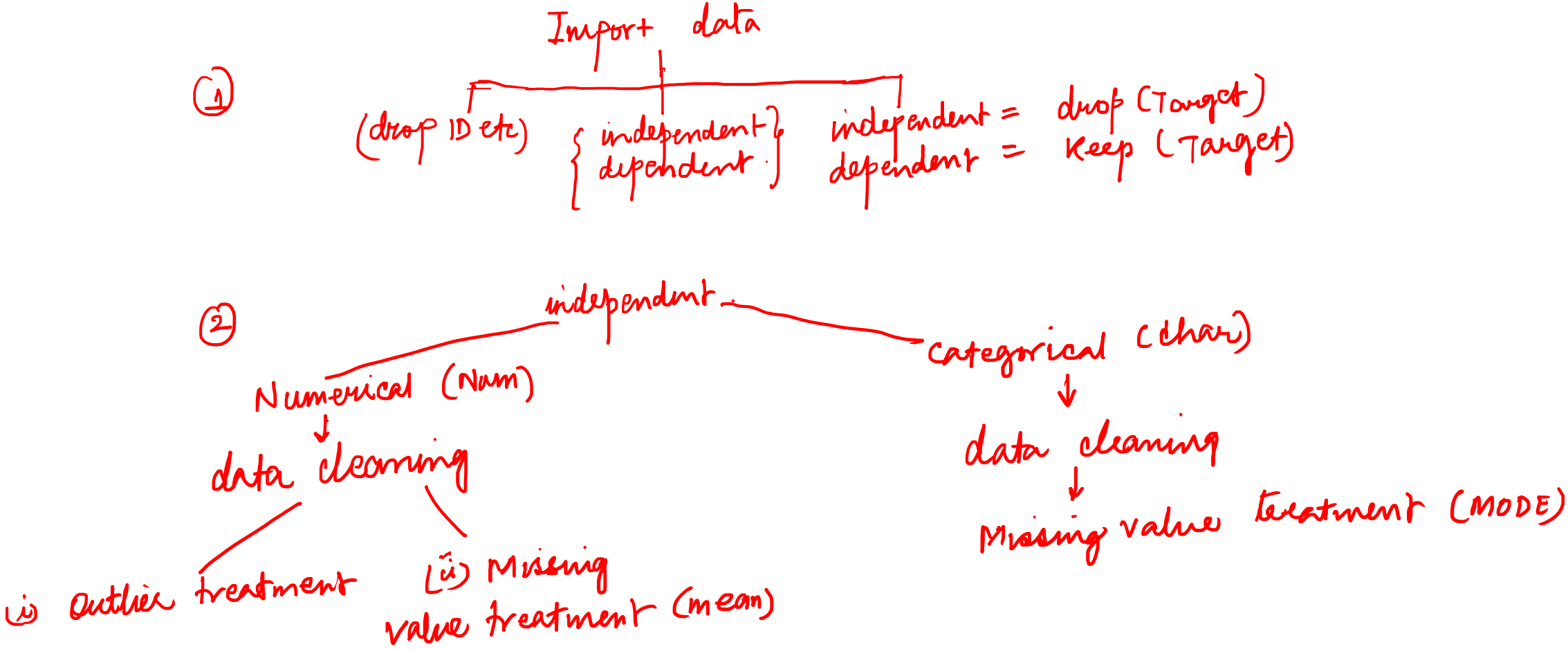
[every rank will represent 10% of the data.
Rank 0 → lowest 10% of ages
Rank 9 → highest 10% of ages]



i) Age discriminates well

ii) just on the basis of age, you could create groups

Python Implementation Flowchart



(Feature selection)

③ Num (Numerical df)

i) Remove all those features that have a 0 variance

ii) we will use the class called KBinsDiscretizer

iii) develop Bivariate graphs → {Remove non predictive features}

iv) use (SelectKBest) → select the final set of features from (num)

④ char (categorical df)

i) develop Bivariate graphs → (Remove non predictive features)

ii) select K Best → select the final set of features (char)

(Applying Algorithms)

Statistical
{Logistic}

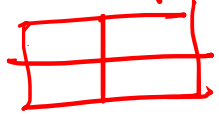
$\log\left(\frac{y}{1-y}\right) = f(x)$

ML (Tree Based)

{Decision Tree
Random forest
GBM}

(predict 50% cut off)

Evaluation

✓ i) Accuracy, Precision etc
✓ ii)  confusion matrix

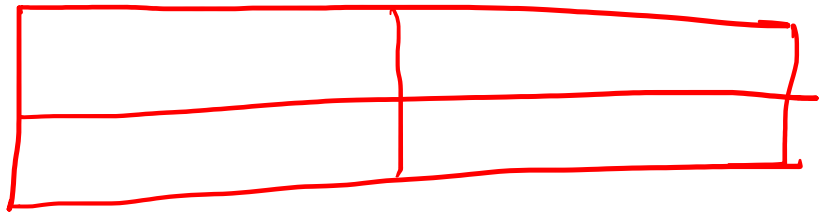
[Probability of Y = 0/1]

i i i) LORENZ CURVE

} 10 Ranks / Deciles in ascending order.

{ Total churners / Total customer = x% } ↑↑ x%

Low value ---- --- -> High value



Top 3 (Based on prob of Y=1)

Bottom 7 "

- (A) we will identify the audience for targeting
- (B) we will identify the treatments to apply (what to offer)
- (C) we will chalk out a prioritization from Low → High priority for targeting

create recommendations based on the model outcome —— ☆☆☆.