Cross Validation

(a) We cant just train our ML model on any dataset & expect it to work on real data

(b) There must be some assurance that data will work

(c) For this Testing is required

(d) Cross Validation is a stastical method to estimate the performance of a model

(e) Methods of cross validation

i) Non Exaustive → do not compute all ways of splitting the dataset

ii) Exaustive → compute all ways of splitting dataset

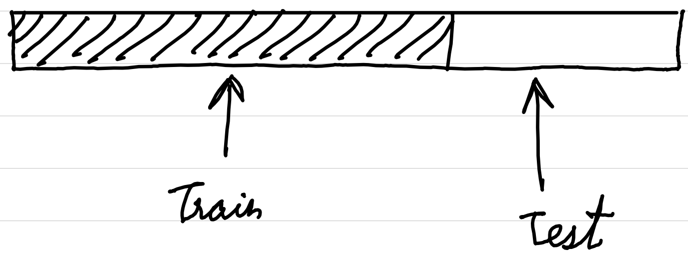
Cross Validation

- Excaustive
  - Leave P out
  - Leave 1 out
- Non Excaustive
  - Holdout Model
  - Random Subsampling
  - K fold cross validation

Splitting a dataset

(a) In order to check if a model works on new data or not, we split it into two parts.

(b)



for training, use the training dataset

for testing use the testing dataset

dont use testing dataset anywhere in training the model

(c) Measurement of accuracy etc will be done on testing dataset

(d) After evaluating the model, train the model on entire dataset with same hyperparameters

Holdout Model

ⓐ Holdout Model randomly shuffles the data, splits it into training & testing

ⓑ Drawback → Suffers from high Variance

Not certain which data point will end up in testing dataset

Result might vary & be entirely different for different sets

ⓒ If training is large, model will be good,
but that means testing is small so estimation is less reliable

ⓓ However, this model will require the model to be trained only once, hence fastest

Shuffle data
↓
Test train split
↓
Train
↓
Calculate accuracy
↓
Hyperparameter tuning
↓
Train on entire dataset
↓
output

Repeated Random Subsampling

(a) It is variation of holdout Model

(b) Instead of performing holdout Model once, do it many times.

(c) Accuracy of model is average of all other accuracies

Shuffle data
↓
split into Test & Train
↓
Train Model
↓
Calculate Accuracy

k times

↓
Total Accuracy of Model is Average of all Accuracies
↓
Hyperparameter tuning

until desired accuracy

↓
Train on entire dataset
↓
output

(d) This requires model to be trained multiple times before hyperparameter tuning

Hence computationally expensive

(e) Also known as Monte Carlo Cross Validation and shuffle split cross validation

k fold Cross Validation

(a) Random subsampling does not have control over number of times each tuple is used for training or testing

(b) What if one data point is used again & again?

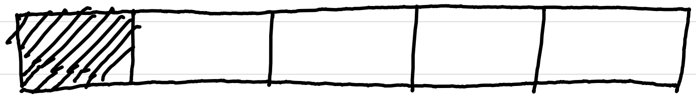
(c) Random subsampling is random we need more concrete

(d) So rather than making every data point random chance, we give every datapoint equal chance

(e) Divide dataset into random k folds

(f) Use each fold once for testing

(g) So every datapoint is excluded for training once.



k folds

eg for 4 folds

iter 1	Test	Train	Train	Train
iter 2	Train	Test	Train	Train
iter 3	Train	Train	Test	Train
iter 4	Train	Train	Train	Test

(h) Accuracy of the Model is the average of all k accuracies

Stratified K-fold Cross Validation

(a) In k fold crossvalidation there are chances that we might get highly unbalanced folds

(b) This leads to increasing bias

(c) eg in one fold all +ve values are present & in second one all -ve values occur

(d) To avoid this stratification is used

(e) Stratification is a process of rearranging the data so as to ensure that each fold is a good representative of the whole

(f) Ensure that every fold has an equal amount of every class

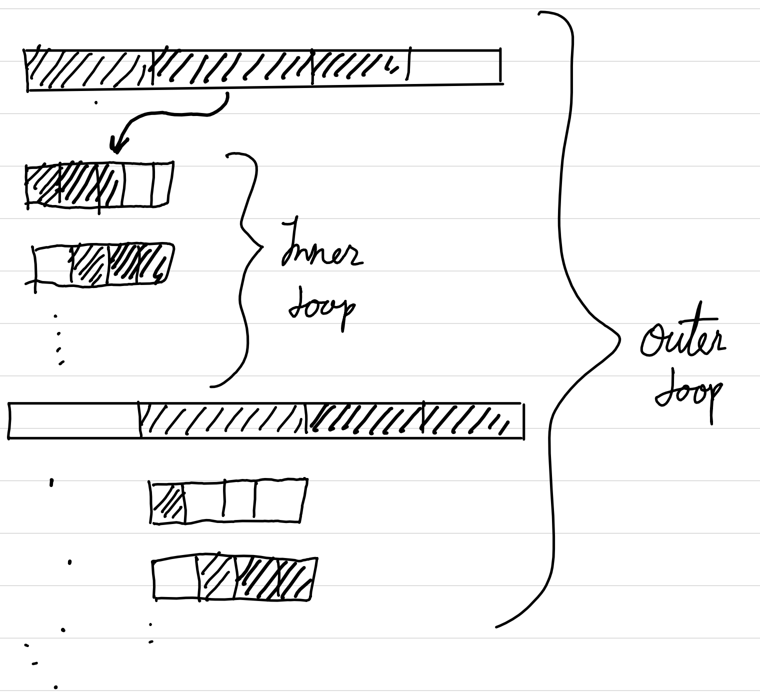
Example → if there are 40% samples of cat & 60% samples of Dog, in dataset each fold must contain same proportion 40% cat & 60% dog

Nested k-fold

Nested k-fold cross validation provides more reliable performance of model when hyperparameter tuning is involved

Even the hyperparameters are included in the model itself.

Extra layer of cross validation to better estimate hyperparameters.



Outer loop (n folds)

1. split data into n folds
2. Iteration 1 : Use fold 1 of n for testing and others for training

   Inner loop (k folds)

   1. split the training dataset to k folds
   2. Train Model & validate
   3. Average Accuracy

3. Calculate Hyperparameters based on the accuracy. choose best hyper parameters
4. Train the entire fold (n-1 of n) using best hyper parameters
5. Test on the fold 1 of n
6. Go to iteration 2 ... n
7. Train on all dataset
8. Model is ready

This takes huge time complexity

If $n * k$ models are fit & evaluated as part of traditional cross validation then it is increased to $k * n * k$

Leave P out

ⓐ Leaves P datapoints out of training data

ⓑ For all possible combinations of P find the effectiveness

ⓒ for modarelly large P, it becomes computationally infeasible

Leave 1 out

ⓐ P=1

ⓑ Extreme case of K fold crossvalidation where size of a fold is 1

ⓒ Useful for small models

ⓓ No of models = No of samples

ⓔ Used when accurate estimate of model performance (dont confuse with accurate Model) is required

Evaluation

ⓐ ML algorithm has 3 types of errors

i) Bias Error
ii) Variance Error

} Reducible

iii) Noise – irreducible?

ⓑ Goal of improving ML algorithm is to reduce the errors

Bias

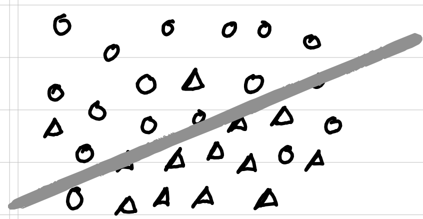
(a) Bias is the difference between predicted value & absolute value

(b) High bias means that the model has not learnt well

Low bias means model is matching well with the dataset

(c) This is called underfitting

(d) This means model is very simplistic



Small dataset can cause high bias

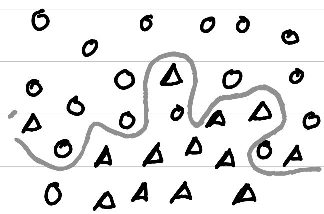
(e) Bias is a measure how close the model can capture the mapping function between inputs & outputs

Variance

ⓐ Variance is error introduced due to complex algorithm

ⓑ This means model learns the noise along with good parts

ⓒ Model performs well on training data ie training accuracy is high



ⓓ But it fails on testing or unknown data

ⓔ It is like rote learning a question bank without understanding anything

ⓕ This is called overfitting

ⓖ This makes model rigid

ⓗ Variance is the amount by which the model would change if we estimated it using a different dataset

The tradeoff

(a) A Model with high bias error underfits data & makes simplistic assumptions

(b) Model with high variance overfits data and learns too much from it

(c) A good model needs balance between the two

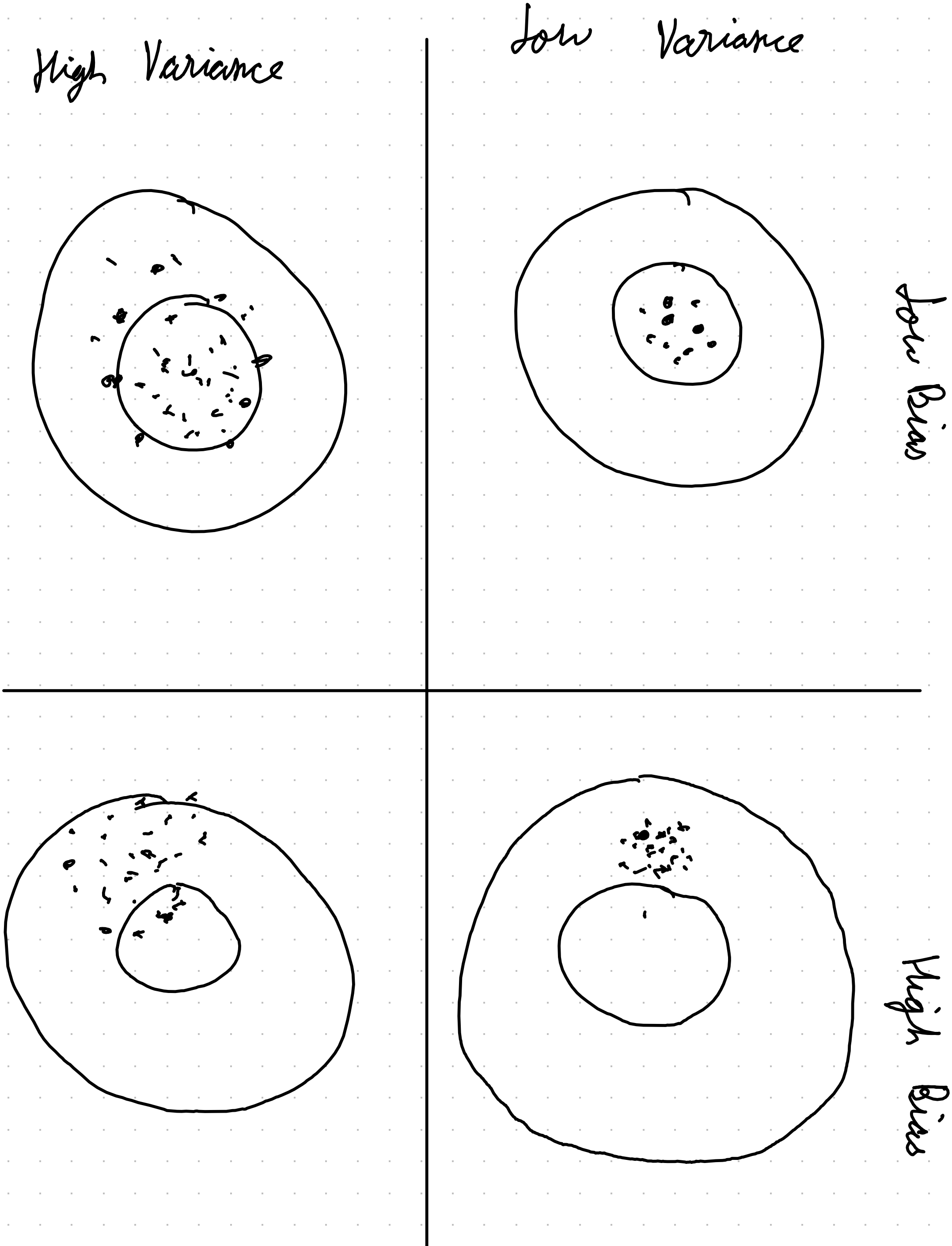
(d) In the ideal case, bias & variance must be low

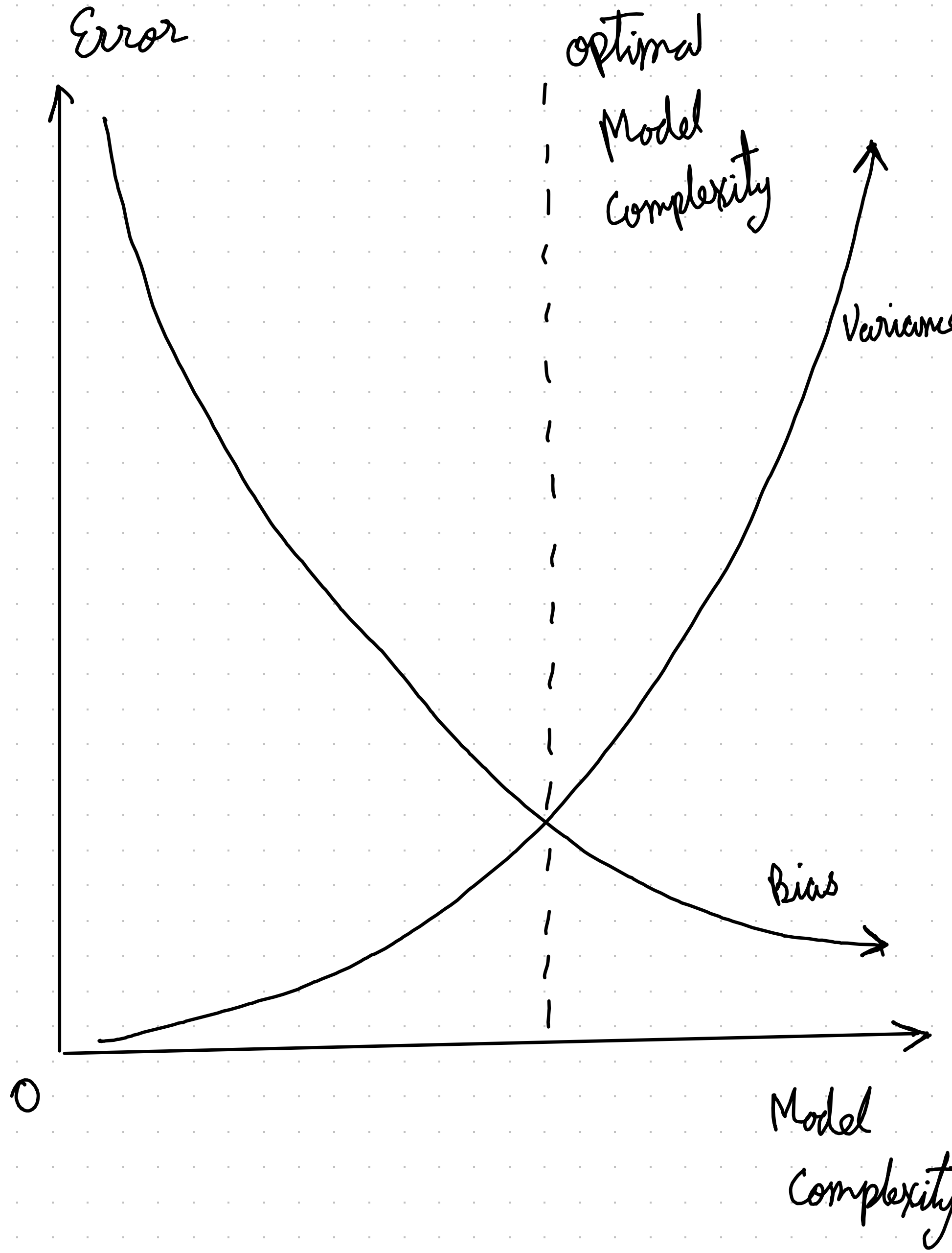
(e) But decreasing bias will increase variance & vice versa

(f) That means your model will either be dumb or oversmart

(g) The key is to stop when golden mean is reached

(h) Some models naturally have high bias or high variance which needs to be adjusted with help of hyperparameters or regularization techniques.


High Variance
Low Variance
Low Bias
High Bias


Error
optimal
Model
Complexity
Variance
Bias
0
Model
Complexity

Confusion Matrix

Used to evaluate the Model

Predicted class		Actual class	
		cat	noncat
	cat	TP	FP
	Non cat	FN	TN

(a) Accuracy $= \frac{TP + TN}{Total} \times 100$ i.e. $\frac{correct}{Total} \times 100$

(b) Precision $= \frac{TP}{TP + FP}$ (P for Precision)

$$= \frac{\text{samples correctly predicted of A}}{\text{samples predicted of A}}$$

(c) Recall $= \frac{TP}{TP + FN}$ (also called sensativity)

$$= \frac{\text{Samples correctly predicted of A}}{\text{Samples of A}}$$

(d) F1 score → Harmonic Mean of precision & recall

$$= \frac{2 \times \text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}}$$

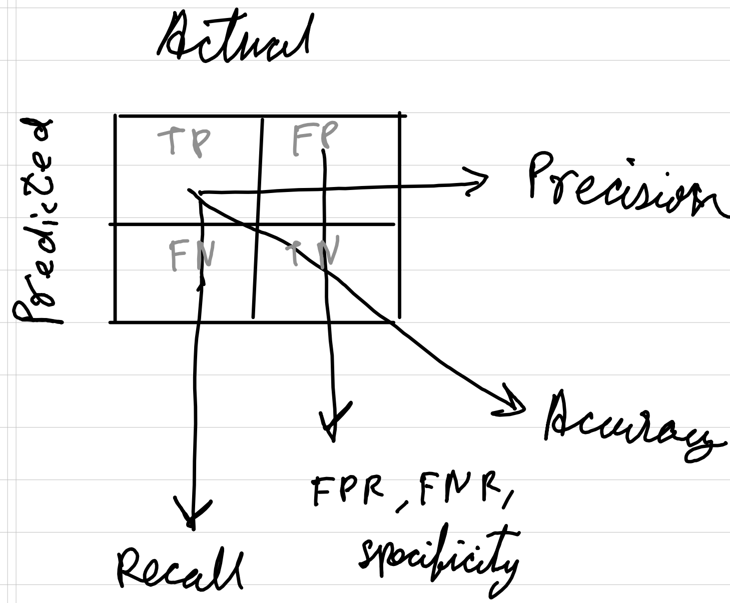
(e) Specificity $= \frac{TN}{TN + FP}$

i.e. proportion of negative data points that are correctly considered as negative of A

(f) False Positive rate $(FPR) = \frac{FP}{TN + FP}$

Proportion of negative data points that are mistakenly considered positive with respect to all negative data points

(g) False Negative Rate $(FNR) = \frac{FN}{TN + FP}$



Confusion Mat: For Multiple classes

TN	FN	TN
FP	TP	FP
TN	FN	TN

FP is also called type I error

FN is called type II error

Q1

Predicted ↓ / Actual →	cat	Noncat
cat	90	60
Noncat	10	940

$$\text{Accuracy} = \frac{90+940}{10+90+60+940} = 93.6\%$$

$$\text{Precision cat} = \frac{90}{90+60} = 0.6$$

$$\text{Precision noncat} = \frac{940}{940+10} = 0.989$$

Precision of cat is very poor

Accuracy is not a good measure when the dataset is not balanced

$$\text{Recall}_{\text{cat}} = \frac{90}{90+10} = 0.9 = 90\%$$

$$\text{Recall}_{\text{non cat}} = \frac{940}{940+60} = 94\%$$

F1 score = 72%

$$\text{Specificity}_{\text{cat}} = \frac{940}{940+60} = 94\%$$

$$\text{FPR}_{\text{cat}} = \frac{60}{940+60} = 6\%$$

$$\text{FNR}_{\text{cat}} = \frac{10}{940+60} = 1\%$$

Precision Recall tradeoff

ⓐ If precision is high, recall is dropped

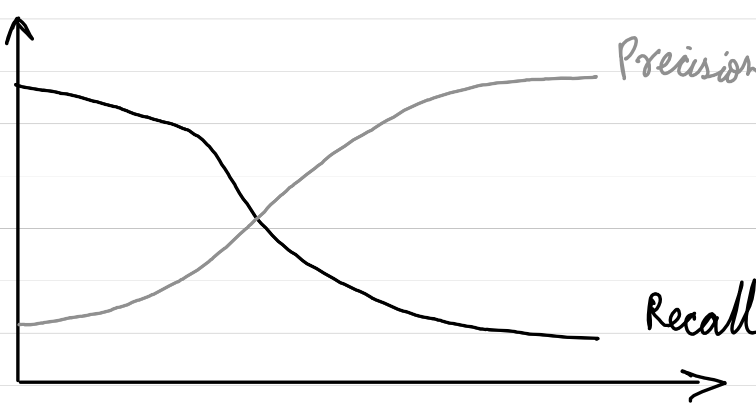
ⓑ Hence to evaluate the model F1 score is used

ⓒ eg if
Precision = 0
Recall = 1

$F_1 = 0$

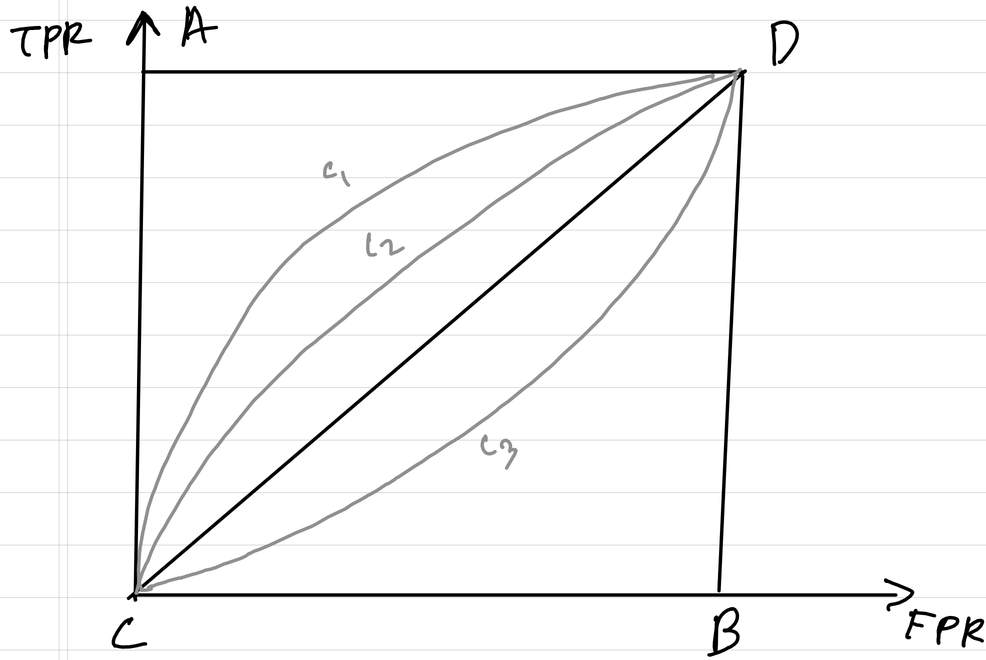
Hence Harmoni mean is taken

ⓓ Higher F1 score the better



ROC curves

Reciever operator characteristic



A → Perfect
B → Worst (flip to Make perfect)
C → Ultra conservative
D → ultra libral

Curves with larges area under curve are better

$c_1 > c_2 > c_3$