Measures of ML model Performance

→ Confusion Matrix →

y actual \ $\hat{y}$ predicted	0	1
0	TN	FP
1	FN	TP

Accuracy rate = $\frac{\text{Correct}}{\text{Total}}$

$$\boxed{AR = \frac{TN + TP}{\text{Total}}}$$

A confusion matrix is predicted result on a classification problem.

Error Rate = $\frac{\text{Wrong}}{\text{Total}}$

$$\boxed{ER = \frac{FP + FN}{\text{Total}}}$$

(TP) True Positive = y actual is positive, & predicted $\hat{y}$ to be positive.

(FN) False Negative = y is positive & $\hat{y}$ is Negative

(TN) True Negative = y is Negative & $\hat{y}$ is to be Negat

(FP) False Positive = y is Negative & $\hat{y}$ is Positive.

Recall

Out of all the +ve samples, what fraction did my classifier pick up.

The total Number of correctly classified positive examples divided to the total No of positive examples.

$$\boxed{\text{Recall} = \frac{TP}{TP + FN}}$$

Precision

Out of all the samples the classifier labelled as +ve, what fraction where correct.

Scanned by CamScanner

Page No.
Date

The total No of correctly classified positive examples by total number of predicted positive examples.

$$\boxed{\text{precision} = \frac{TP}{TP + FP}}$$

High recall, low precision –
most of the positive examples are correctly recognized (low FN) but (high FP)

low recall, high precision –
we miss lots of positive examples (high FN) low (FP).

~~Recall~~ =

n = 165	predicted No	predicted Yes	
Actual No	TN = 50	FP = 10	60
Actual Yes	FN = 5	TP = 100	105
	55	110	

$$\text{Recall} = \frac{TP}{\text{Actual yes}} = \frac{TP}{TP+FN} = \frac{100}{100+5} = 0.95$$

$$\text{precision} = \frac{TP}{\text{predicted yes}} = \frac{TP}{TP+FP} = \frac{100}{100+10} = 0.91$$

Scanned by CamScanner

Precision Recall Trade off

This is a fundamental ~~model trade~~-off betn precision & recall. In our model with high precision (most or all of the fish caught were red) & low recall (we missed a lot of red fish)

In our model with high recall (we caught most of the red fish), we had low precision (also caught lot of blue fish).

F1 score (f measure)

we can combine precision & recall to have an overall score that predicts how well our model is performing.

$$\boxed{F = 2 \cdot \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}}$$

when we taking Arithmatic mean, it would have ~~43%~~ worst possible outcomes.

with harmonic mean, the F1 measures is good

If you have high F1 score, you need to both have a high Precision & recall.

Accuracy Paradox -

Accuracy as freedom from mistake or error. a piece of information is accurate if it exactly represen what is being discussed.

pradox is a statement that is seemingly Contradictory or opposite to common sense

simple model give high ~~performance~~ level of accuracy. which is went something wrong.

CAP Curve

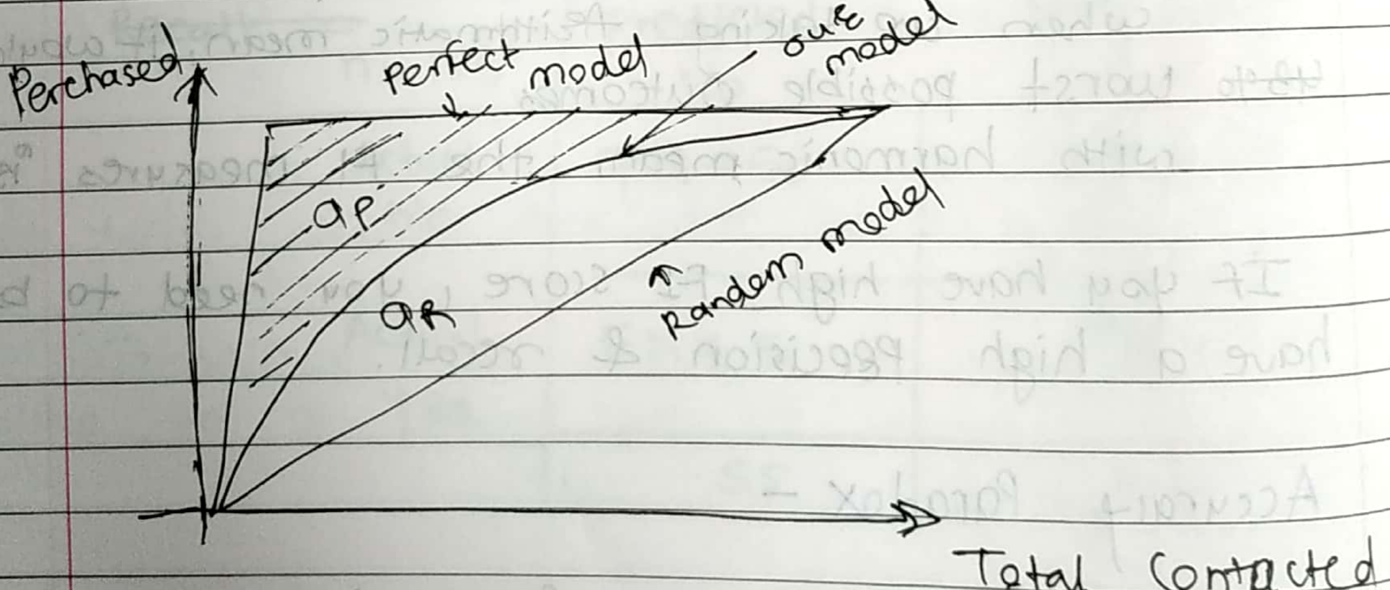
Cumulative Accuracy Profile.

ROC curve & CAP curve are great for classification problem.

CAP can be used to evaluate a model by comparing the curve to perfect CAP in which maximum rate of elements meeting the ~~problem~~ ~~probability~~ property is achieved directly & the random CAP in which the elements meeting property are distributed equally.

A Good model will have a CAP betn the perfect cap & random CAP with a better model tending to perfect CAP.

AR is defined as the ratio of the area betn the model CAP & random CAP.

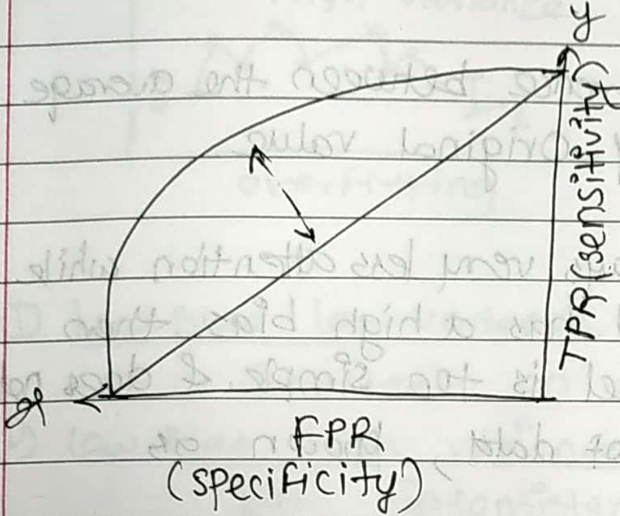


$$\text{Accuracy ratio} = \frac{a_P}{a_R} \quad [\text{betn } 0 \text{ \& } 1]$$

1 is better, 0 worst

AUC or ROC Curve

ROC curve are mainly used to Binary classifier performance, classifier with two possible outcomes



$$TPR = \frac{TP}{\text{all positive}}$$

$$FPR = \frac{FP}{\text{all Negative}}$$

[Range - 0 to 1]

ROC curves are useful even if your predicted probabilities are not "properly calibrated".

ROC tells us (what) how many mistakes are (FP) we making to find True positive (TP)

FPR = (1 - specificity)

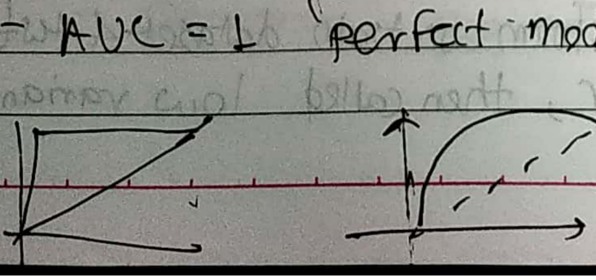
we have to find as less as less mistake to find TP ie make (0% error) (least FP rate).

Good model - AUC is large.

AUC

We want area under curve to be far away from straight line.

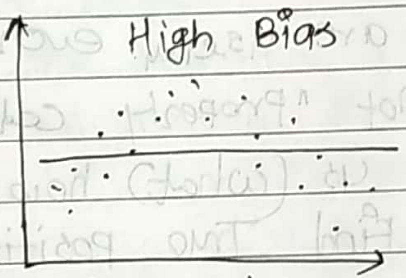
- ROC curve give us whole area under curve
- AUC = 1 perfect model



Bias - Variance Trade off -
[concept of overfitting & under fitting]
predict a model which has minimum (MSE) mean Square Error

Bias → It is a difference between the average of predicted value by original value.

High Bias → model plays very less attention while learning, when model has a high bias then it implies that the model is too simple & does not capture the complexity of data, known as Underfitting



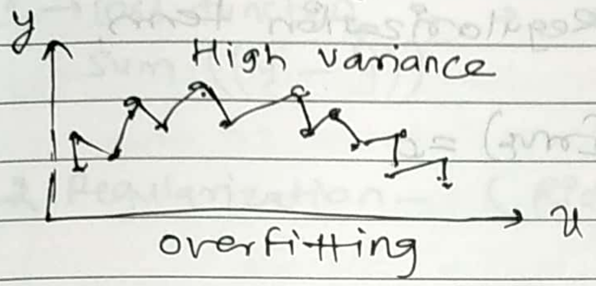
Underfitting

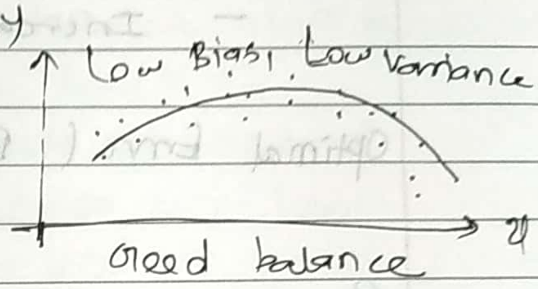
Low Bias → model plays better attention while learning which might lead to overfitting

Variance -
When a model perform well on data it is trained on but performs poorly when we use other similar dataset like test dataset or validation datasets.
It tells how scattered ~~value~~ our predicted value are from the actual value.

Low Variance
When model perform on train dataset & test/validation dataset is quite similar, then called low variance.

High variance - The model becomes very flexible & tunes itself to the data points of training set. (overfitting)





① Low Bias low variance - models are accurate & Consistent on average

② Low Bias High variance - models are somewhat accurate but inconsistant.

③ High Bias low variance - constant but inaccurate

④ High Bias High variance - inaccurate & inconstant.

Find when have high Bias & variance

High Bias →
High training Error, validation or test error is also high.

High variance →
Low training Error, high validation error or test error

How to fix

Underfitting is due to a simple model.
- add more input feature
- Add more complexity by Introducing polynomial feature
- Decrease Regularization term.

Overfitting (high variance)
- Getting more training data
- Reduce input features
- Increase Regularization term.

Optimal Error (Bays Error) = 0

Regularization
- It helps overcoming overfitting issue.
- It is called regularization as it helps keeping the parameters regular or normal.
- Generally we do not want huge weights
- A small change in weight make large difference in the target variable.
- 0 weight for feature that are not very imp.
- Not too much weight any feature.

$$J(\theta) = \frac{1}{2m} \cdot \Big[\underbrace{\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{\text{cost}} + \underbrace{\lambda}_{\text{Regularization param}} \cdot \sum_{j=1}^{n} \theta_j^2 \Big]$$

L1 & L2 Regularization -

Regularization Techniques →
1. L1 & L2 regularization
2. Dropout
3. Data augmentation
4. Early stopping.

Date

L1 ~~loss~~ function → when we are trying to compress our model
$sum(|y-\hat{y}|)$

L2 → loss function → otherwise
$sum((y-\hat{y}))^2$

L2 Regularization – (Ridge)

~~$\omega^* = \underset{\omega}{argmin}$~~ $\sum_{i}$

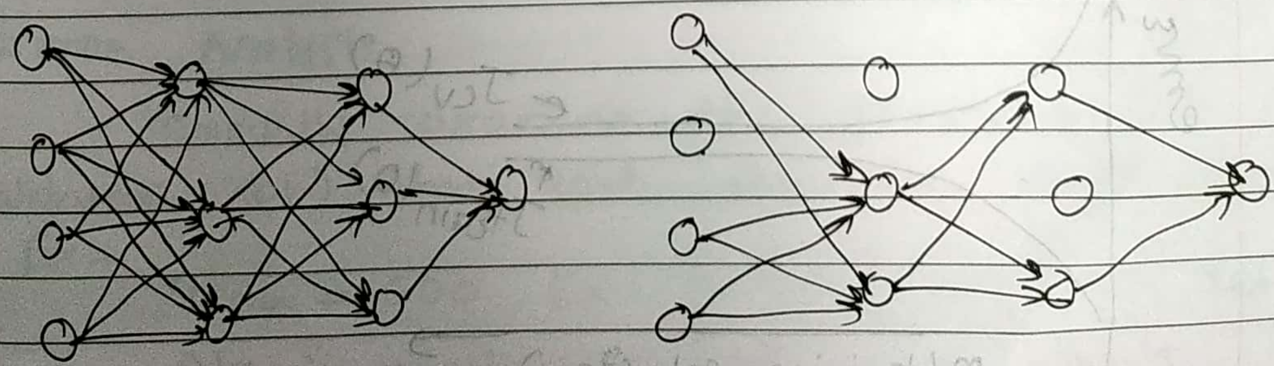
$$\text{cost function} = \text{loss} + \frac{\lambda}{2m} * \sum \|\omega\|^2$$

L1 Regularization – (Lasso)

$$\text{cost function} = \text{loss} + \frac{\lambda}{2m} * \sum \|\omega\|$$

Usage
- To reduce over fitting
- To select imp features so that prediction accuracy good
- finding optimal weights

Dropout
most frequently used Regularization Technique in Deep learning.



with every iteration we randomly dropout afew neurons along with their input & output

Data Augmentation

It is used in computer vision / CNN : we just increase the size of our dataset so that our model can generalize easily.

- ↑se size of Training dataset
- rotating img, flipping, scaling, shifting

Early stopping

It is kind of cross validation strategy where we keep one part of training set as the validation set, when we see performance on the validation set is getting worse, we immediately stop the training on model., this called Early stopping

Learning Curves

It shows the relationship between training set size & your chooesen evaluation metrics on your training & validation sets. They can be an extermely useful tool when diagnosing your model performance, they can tell you wether your model is suffering from bias or variance.

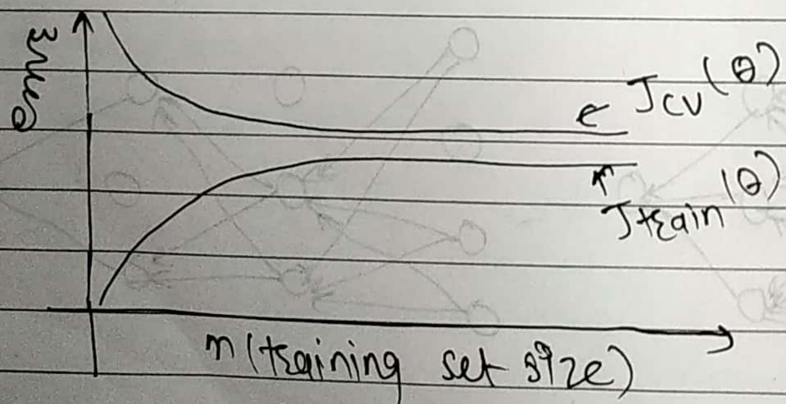


fig. high Bias model

If your curve looking like this then, your model suffering from high bias. Both training & validation error is high & doesn't seem to improve with more training example.
model is performing bad for both Training & validation set. (underfit data → high bias)

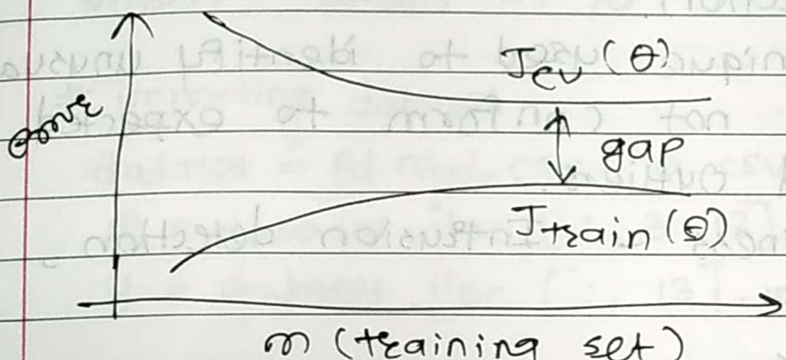


fig. high-variance model

If learning curve, looks like this then then model have high variance problem.
- Validation error is much higher than Training Error (overfitting data)

Validation curve

- plotting score to evaluate models.

To validate a model we need a scoring function.
If the training score & validation score are both low, estimator will be underfitting. If training score is high & the validation score is low, the estimator is overfitting otherwise, work very well.
A ~~high~~ low training score & high validation score is not possible.

Error Analysis

Manually examine the examples (in cross validation set) that your algorithm made error on.
Ex –
$m_{cv} = 500$ Ex. in cross validation set.
Algorithm misclassified 100 emails.
Manually examin the 100 errors, & categorize them based on:

1) what type of curve it is
2) what cues (features) you think could have helped the algo. classify them correctly.

Anamaly Detection.

It is a technique used to identify unusual patterns that do not conform to expected behaviour, called outliers.

Appl^on in business - Intrusion detection, fraud detection.

1) point Aanamalies -

A single instance of data is anamalous if its too far off from the rest.

Anamaly Detection Techniques

- to identifying irregularities in data is to flag the data points that deviates from common statistical properties of a distribution, including mean, mode, median.

- That deviates by a certain standard deviat from mean.

L1 Regularization	L2 regularization
Computational inefficient on non sparse case	Computational efficient due to having analytical solutions
sparse output	Non sparse output
Built-in feature selection	No feature selection
L1 loss function	L2 loss function
- Robust	Not very robust
unstable sol^on	stable sol^on
possibly multiple sol^on	Always one sol^on