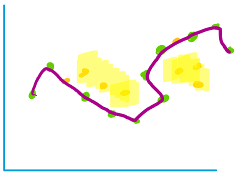
Topics Left

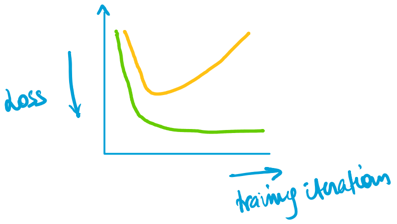
29 May 2022 23:42

- ✓ ① Underfitting vs Overfitting
- ✓ ② Handling missing values.
- ✓ ③ ROSE & SMOTE
- ④ Bagging & Boosting.
- ✓ ⑤ Resume Preparation.

Overfitting (high variance)



* Model performs well on training data but performs poorly on unseen/validation data.



How to handle:

① Increase training data: If training data is less a complex model can remember the training data.

② Decrease model complexity: It's hard for simple model to memorize the training data.

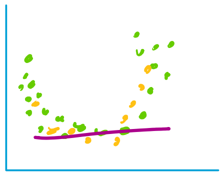
③ Regularization:

L1 regularization,
L2 regularization: Forces the model to keep model weights small by penalizing larger weights

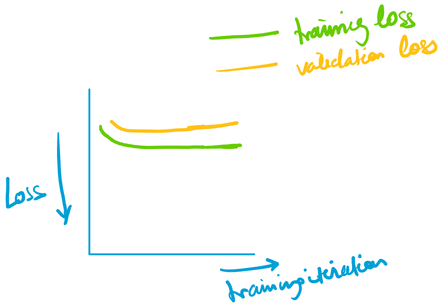
Dropout: Randomly killing neurons while training, this forces model to not be overly dependent on selective neurons/parameters.

Underfitting (high bias)

○ → Training data
○ → Unseen data
○ → learned function



* The model performs poorly on training data.



How to handle:

① Increase model complexity

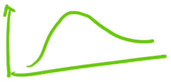
Handling missing Values in Training/Input data:

X_1

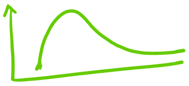
	X_1	X_2	X_3	X_4	Y
1	0.1	yes	2	0.1	0
2	0.15	yes	3	0.15	1
3	0.18	yes	1	4.0	1
4	0.2	yes	3	0.8	0
5	-	no	-	-	
	↓ mean	↓ mode	↓ mean	→ median	

options: mean, median, mode

mean → • when data is numerical & not skewed
• mean is affected by outliers

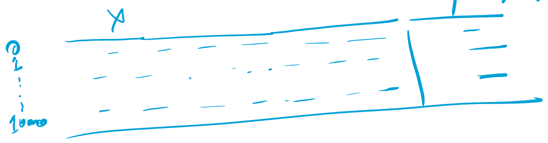


median → when data is numerical & skewed
• median is mildly affected by outliers



mode → when data is categorical & skewed.

Handling unbalanced data set.



Y → possible values → irrelevant, climate, crime, traffic

irrelevant	climate	crime	traffic
↓	↓	↓	↓
800 training samples	100 training samples	60 t.s.	40 t.s.

heavily unbalanced

ROSE → Random over sampling of minority classes.

(numerical & categorical)

climate samples (100) → Random over sampler → 500

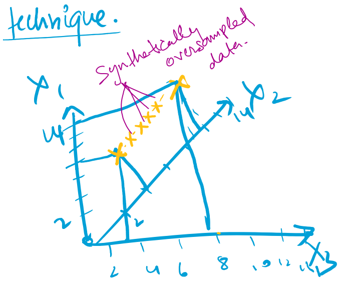
crime (60) → ROSE → 400

climate (40) → ROSE → 400

SMOTE → Synthetic minority oversampling technique.

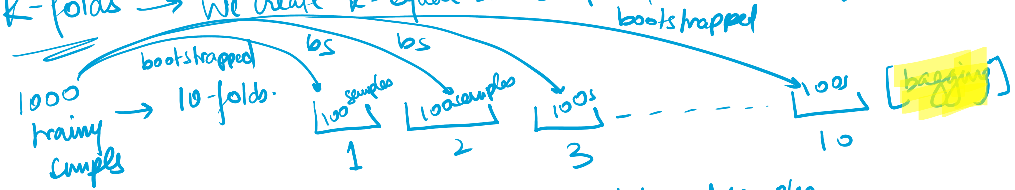
(numerical data)

X_1	X_2	X_3	Y
1	2	1	●
			●
1.5	2.5	1.5	✗
3	4	7	●
2	3	4	✗
8	11	15	●

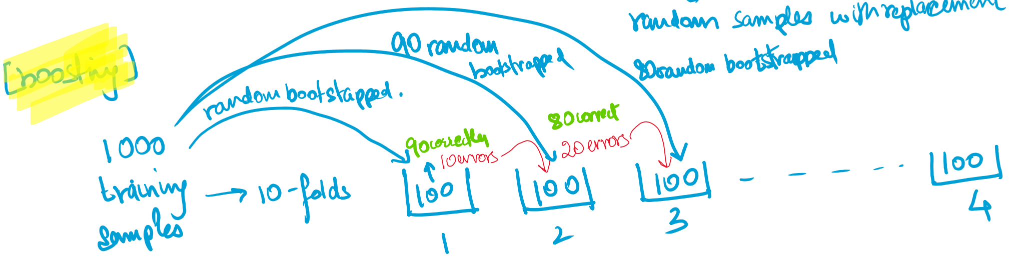


Bagging & Boosting

K-folds → We create k-equal sized samples from total training data.



All buckets have bootstrapped samples
↓
random samples with replacement



Random Forest is a Bagging algorithm → Every tree is trained on a bootstrapped sample.