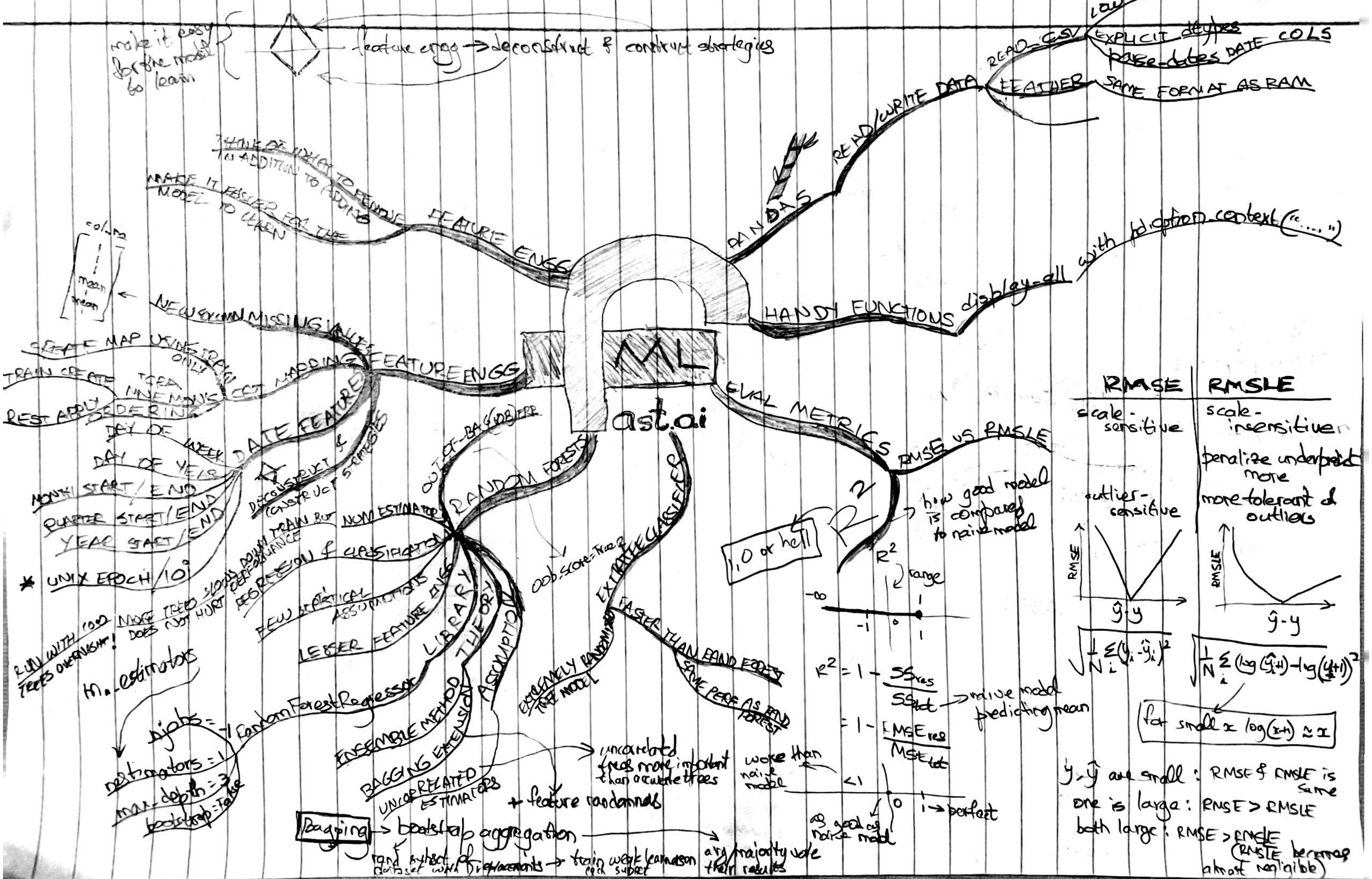
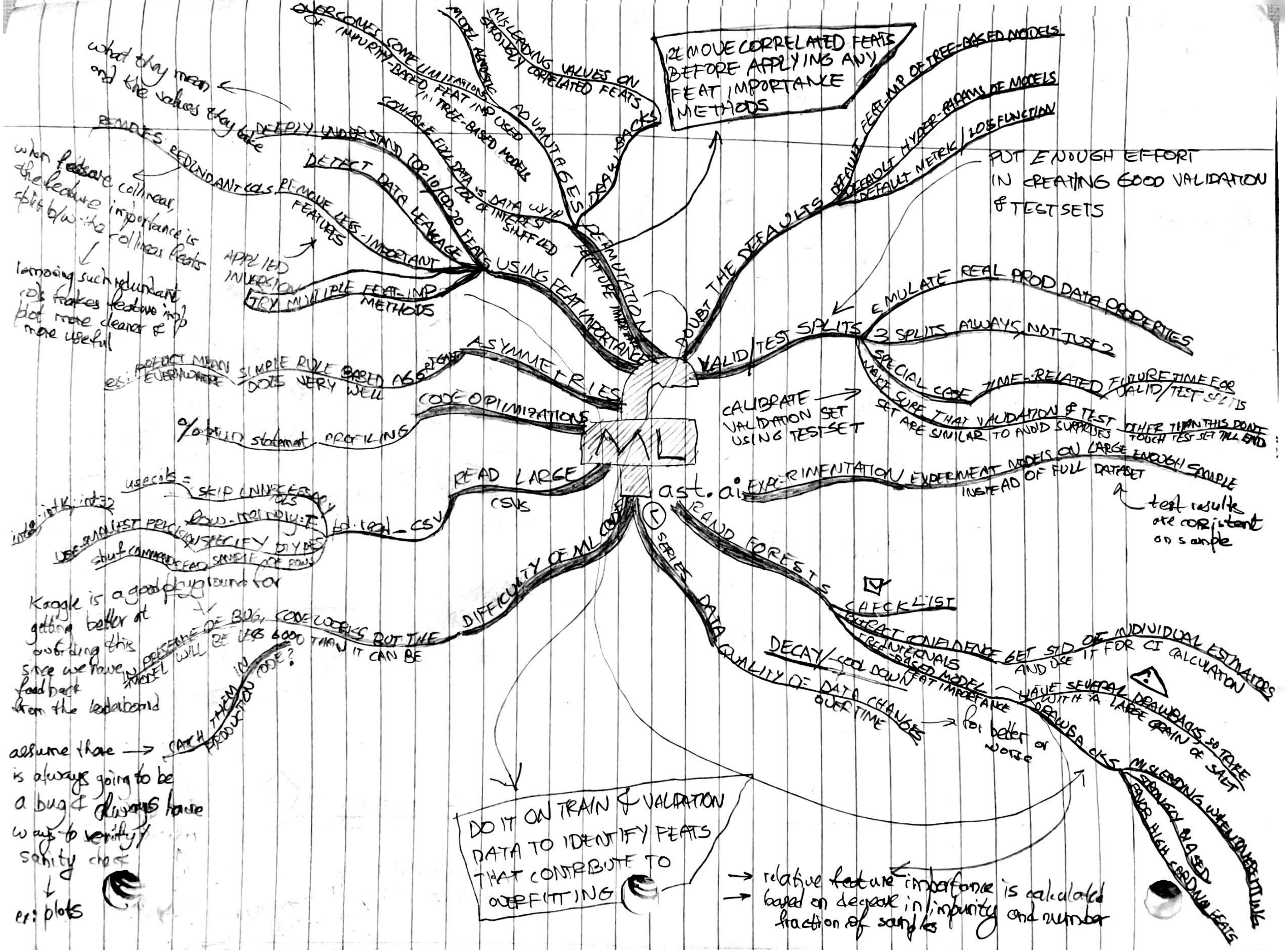
* Epoch : 1.67M → 2023
0.03M Δ FOR 1 YEAR



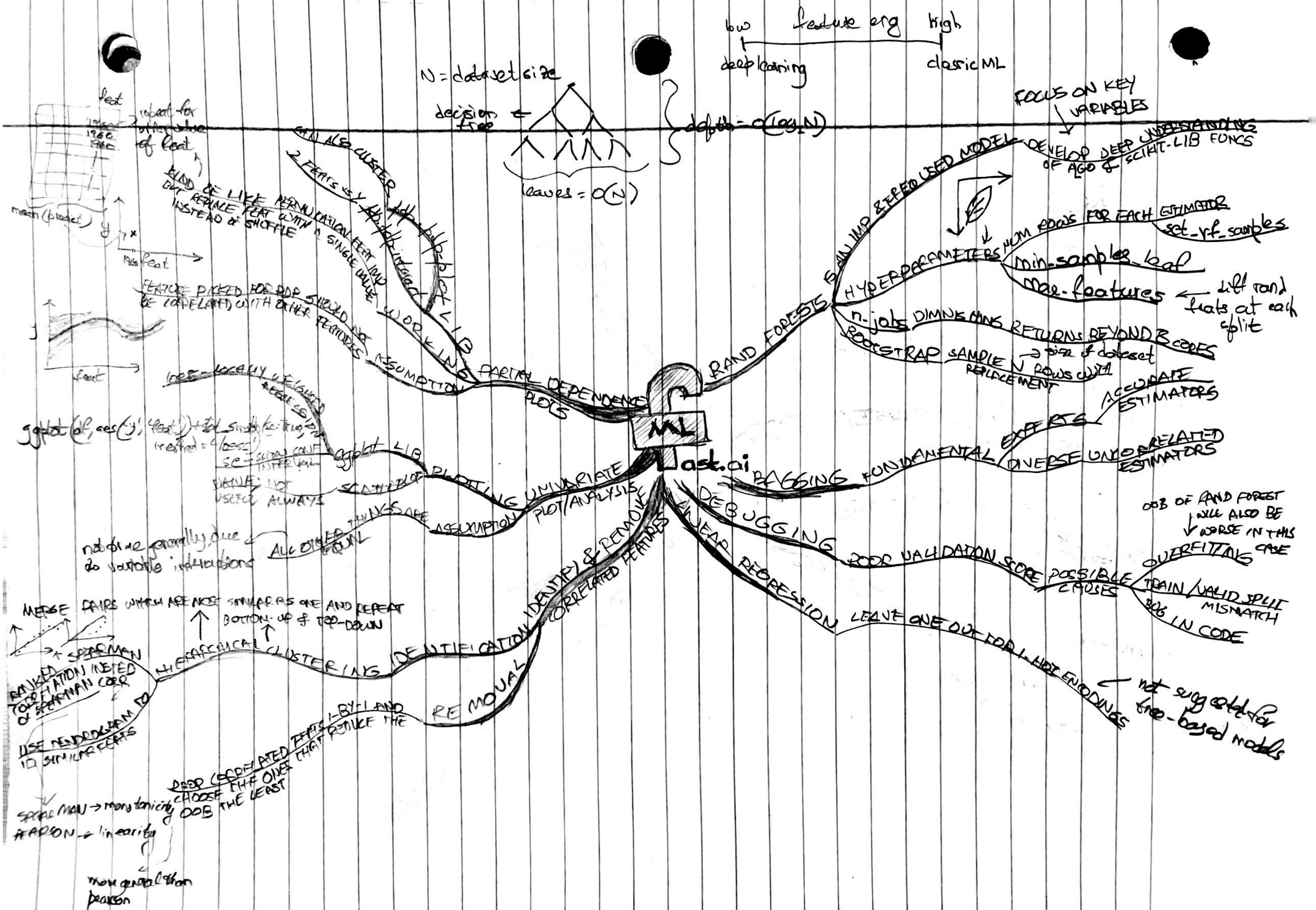
$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{MSE_{res}}{MSE_{tot}}$$

RMSE	RMSLE
scale-sensitive	scale-insensitive
outlier-sensitive	penalize underpredict more
	more tolerant of outliers
$\sqrt{\frac{1}{N}\sum_i (y_i - \hat{y}_i)^2}$	$\sqrt{\frac{1}{N}\sum_i \left(\log(\hat{y}_i+1) - \log(y_i+1)\right)^2}$

$y, \hat{y}$ are small : RMSE & RMSLE is same
one is large : RMSE > RMSLE
both large : RMSE > RMSLE (RMSLE be error almost negligible)



ML
USING FEAT IMPORTANCE
DEEPLY UNDERSTAND TOP-10/TOP-20 FEATS
what they mean and the values they take
REMOVES REDUNDANT COLS
when feature collinear, the feature importance is split b/w the collinear feats
removing such redundant cols makes feature imp plot more clearer & more useful
DETECT DATA LEAKAGE
REMOVE LESS-IMPORTANT FEATURES
TRY MULTIPLE FEAT-IMP METHODS
APPLIED INVERSION
COMPARE FULL DATA vs DATA WITH COL OF INTEREST SHUFFLED
FEATURE IMPORTANCE
PERMUTATION
ADVANTAGES
OVERCOME LIMITATIONS OF IMPURITY-BASED FEAT IMP USED IN TREE-BASED MODELS
MODEL AGNOSTIC
DRAWBACKS
MISLEADING VALUES ON STRONGLY CORRELATED FEATS
REMOVE CORRELATED FEATS BEFORE APPLYING ANY FEAT IMPORTANCE METHODS
DOUBT THE DEFAULTS
DEFAULT FEAT-IMP OF TREE-BASED MODELS
DEFAULT HYPER-PARAMS OF MODELS
DEFAULT METRIC / LOSS FUNCTION
PUT ENOUGH EFFORT IN CREATING GOOD VALIDATION & TEST SETS
VALID/TEST SPLITS
EMULATE REAL PROD DATA PROPERTIES
3 SPLITS ALWAYS, NOT JUST 2
SPECIAL CASE TIME-RELATED
FUTURE TIME FOR VALID/TEST SPLITS
MAKE SURE THAT VALIDATION & TEST SET ARE SIMILAR TO AVOID SURPRISES
OTHER THAN THIS DON'T TOUCH TEST SET TILL END
CALIBRATE VALIDATION SET USING TEST SET
ASYMMETRIES
SIMPLE RULE BASED ALGORITHMS DOES VERY WELL
ex: PREDICT MEAN EVERYWHERE
CODE OPTIMIZATIONS
%prun statement PROFILING
READ LARGE CSVs
pd.read_csv
usecols = SKIP UNNECESSARY COLS
low_memory=F
USE SMALLEST PRECISION / SPECIFY DTYPES
int8, int16, int32
shuf command READ SAMPLE OF ROWS
DIFFICULTY OF ML CODE
IN PRESENCE OF BUG, CODE WORKS BUT THE MODEL WILL BE LESS GOOD THAN IT CAN BE
Kaggle is a good playground for getting better at avoiding this since we have feedback from the leaderboard
CATCH THEM IN PRODUCTION CODE?
assume there is always going to be a bug & always have ways to verify / sanity check
ex: plots
fast.ai
EXPERIMENTATION
EXPERIMENT MODELS ON LARGE ENOUGH SAMPLE INSTEAD OF FULL DATASET
test results are consistent on sample
RAND FORESTS
CHECKLIST
EXTRACT CONFIDENCE INTERVALS
GET STD OF INDIVIDUAL ESTIMATORS AND USE IT FOR CI CALCULATION
TREE-BASED MODEL FEAT IMPORTANCE
HAVE SEVERAL DRAWBACKS, SO TAKE WITH A LARGE GRAIN OF SALT
DRAWBACKS
MISLEADING WITH MULTICOLLINEARITY
STRONGLY BIASED
FAVOR HIGH CARDINAL FEATS
relative feature importance is calculated based on decrease in impurity and number fraction of samples
SERIES
DATA
DECAY / COOL DOWN
QUALITY OF DATA CHANGES OVER TIME
for better or worse
DO IT ON TRAIN & VALIDATION DATA TO IDENTIFY FEATS THAT CONTRIBUTE TO OVERFITTING



ML
fast.ai
low
feature eng
high
deep learning
classic ML
N = dataset size
decision tree
depth = O(log N)
leaves = O(N)
RAND FORESTS
IS AN IMP & FREQ USED MODEL
DEVELOP DEEP UNDERSTANDING OF ALGO & SCIKIT-LIB FUNCS
FOCUS ON KEY VARIABLES
HYPERPARAMETERS
NUM ROWS FOR EACH ESTIMATOR
set_rf_samples
min_samples_leaf
max_features
diff rand feats at each split
n_jobs
DIMINISHING RETURNS BEYOND 8 CORES
BOOTSTRAP SAMPLE N ROWS WITH REPLACEMENT
size of dataset
BAGGING
FUNDAMENTAL
EXPERTS
ACCURATE ESTIMATORS
DIVERSE UNCORRELATED ESTIMATORS
DEBUGGING
POOR VALIDATION SCORE
POSSIBLE CAUSES
OVERFITTING
OOB OF RAND FOREST WILL ALSO BE WORSE IN THIS CASE
TRAIN / VALID SPLIT MISMATCH
BUG IN CODE
LINEAR REGRESSION
LEAVE ONE OUT FOR 1-HOT ENCODINGS
not suggested for tree-based models
PARTIAL DEPENDENCE PLOTS
WORKING
ASSUMPTION
PDP LIB
pdp_plot
pdp_interact
2 FEATS x, y
CAN ALSO CLUSTER
KIND OF LIKE PERMUTATION FEAT IMP BUT REPLACE FEAT WITH A SINGLE VALUE INSTEAD OF SHUFFLE
FEATURE PICKED FOR PDP SHOULD NOT BE CORRELATED WITH OTHER FEATURES
feat
mean (predict)
predict for single value of feat
y
x
feat
PLOTTING UNIVARIATE PLOT/ANALYSIS
ggplot LIB
ggplot(df, aes('y', 'feat')) + stat_smooth(se=True, method='loess')
loess - LOCALLY WEIGHTED REGRESSION
se - SHOW CONF INTERVAL
SCATTERPLOT
VALUE NOT USEFUL ALWAYS
ASSUMPTION
ALL OTHER THINGS ARE EQUAL
not true generally due to variable interactions
IDENTIFY & REMOVE CORRELATED FEATURES
IDENTIFICATION
HIERARCHICAL CLUSTERING
MERGE PAIRS WHICH ARE MOST SIMILAR AS ONE AND REPEAT
BOTTOM-UP & TOP-DOWN
SPEARMAN
RANKED CORRELATION INSTEAD OF SPEARMAN CORR
USE DENDROGRAM TO ID SIMILAR FEATS
SPEARMAN → monotonicity
PEARSON → linearity
more general than pearson
REMOVAL
DROP CORRELATED FEATS 1-BY-1 AND CHOOSE THE ONES THAT REDUCE THE OOB THE LEAST