ML Cheat Sheet Notes

(S1)

① ML → A subfield of AI, we build model which make prediction Based on data

② Types → supervised (classification, Regression) :- here we have Both input and output columns

→ unsupervised (Anomoly Detection, Clustering, Associate Based learning, Dimensionality Reduction) :- we have only input column

→ Reinforcement learning → No input, output column, self driving car

→ semi-supervised learing → google Photos.

③ steps → ① Frame the Problem, ② Gather data ③ data Preprocessing ④ EDA ⑤ Feature selection and Engineering ⑥ Model development ⑦ model deployment ⑧ Testing ⑨ optimization.

(S6) Regression metrix

① Loss function - MSE, MSA, RMSE, Evalution = R2 score, Adj R2

② $MSE = \frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2$, Not Robust to outliers

③ $RMSE = \sqrt{MSE}$, Not Robust to outlier

④ $MAE = \frac{1}{n}\sum_{i=1}^{n}|y-\hat{y}|$ Robust to outlier, Not differentiable at 0

⑤ R^2 score → Also know as coefficient of Determination and goodness of fit, It compare regression model with the mean,

Formula = 1 − (Sum of squared error of model) / (Sum of squared error of mean)

→ If we add irrelevent column the value of R^2 score increase so we use adjusted R^2 score,

→ value lie in between 0 to 1

⑥ Adjusted R^2 score

$$F = 1 - \frac{(1-R^2)(n-1)}{[n-1-k]}$$

n = number of row

k = number of columns

(57) classification metrix

① Recall, Precision, F1 score, accuracy, confusion metrix

② Accuracy ⇒ Formula = $\frac{\text{Total correct prediction}}{\text{Total Prediction}}$

⇒ It do not tell which type of error

⇒ Not accurate for imbalanced data set

⇒ what accuracy is good → Depend on problem

③ confusion matrix

Actual / Prediction

TP	FN
FP	TN

Type 1 error = FP

Type 2 error = FN

FP = model Falsely Predicting Positive ⇒ actual = Neg, model = Positive

FN = model Falsely Predicting Negative ⇒ actual = Pos, model = Negative

Type 1 = Precision = Email spam classifier

Type 2 = Recall = Cancer detection

Type 2 + Type 1 = F1 score

Multiclass → like cat, dog, rat

Actual \ Prediction	cat	dog	rat
cat	1	3	5
dog	2	4	6
rat	7	8	1

④ Precision

Out of all Positive Prediction, How many truely Belong there

F = TP / (TP + FP), varient → weighted and macro (for multi class system)

⑤ Recall

Out of all positive cases How many get Predicted correctly

F = TP / (TP + FN) → varient → weighte, macro

⑥ F1 score → when Both T1 and T2 error important, varient → weighted macro.

Formula

For class → 1, 2, 3

macro precision = (P1 + P2 + P3) / 3

weighted Precision = ____

(58) Linear regression

① Types - Single, multiple, Polynomical

② In Sklearn, it is implemented in 2 ways

(i) ~~least square~~ Ordinary least square → LinearRegression()

(ii) Gradient decent → SGDRegressor()

③ Assumption → linear realtion btw input and output

→ No multicolinearity

→ residual normally distributed

(4) Gradient Decent → Optimization Technique, if you give a differentiable function to it, it will return minima of it.

(5) Bias → The Inability of ML Algo to capture the Patter of data, low captured Pattern will lead to more Bias,

(6) ~~Difference~~ Varience → Difference in fit in Different Datasets

(7) Bias varience trade off
→ Generally you have either High Bias with low varience, or low Bias with high varience
low Bias + High varience = Overfitting
High Bias + low varience = underfitting
3 ways to deal with it → Regularization, Boosting, Bagging

(8) Regularization
→ Adding a term in loss function to reduce Overfitting
→ L2 (Ridge), L1 (Lasso), Elastic Net
→ Ridge regularization (L2) → (i) Apply when you know all columns are important, term = $\lambda \sum_{i=1}^{n} m^2$ (m = slop, α = HP $(0, \infty)$)
→ Lasso Regularization (L1) → (i) Apply when you know all columns are not important, term = $\lambda \sum_{i=1}^{n} |w|$
→ Elastic Net Regularization → Apply when you dont know are these columns important or not. Combine lasso + Ridge, $a(w)^2 + b(w)$, a, b → HP
High value of α → High Bias and low variance and model underfit and vice-versa

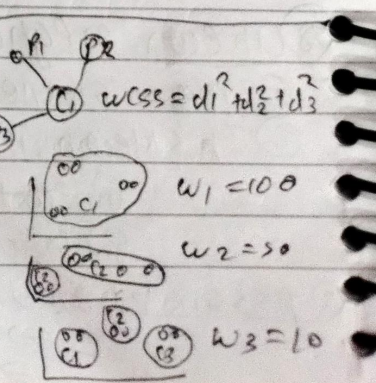
(9) Logistic Regression → Here Polynomial works and softmax
Perceptron trick, sklearn implementation uses Sigmoid function

(10) Decision Tree → Big structure of nested if-else condition, use Hyper Planes which run parallel to any one of the axes to cut coordinate system into Hyper cuboids, used for Regression Problem called → CART (Classification and Regression Tree), overfit a lot

(11) Hyperparametertunning → 2 ways GridSearchCV, RandomSearchCV,

(12) K mean clustering →
Algo → Find number of cluster by Elbow method (k)
→ Assign n random centroid
→
Make clusters Based on centroid, and Point closer to the centroids.
→ Assign centroid again in those clusters
→ if those centroid same as Previous then Finish, else

Elbow method
① From 2 to like 10 or 100 we clarulate wcss, and plot the wcss over graph, and the elbow Point is the number of clusters
② with in cluster ~~sum~~ Sum of squared distance = wcss



(S11) Ensemble learning

(i) wisdom of the crowd, (ii) Types (Voting, Bagging → RF, Boosting, Stacking) (iii) Robust, accurate, low bais, variance

Ada boosting, GD, xgboost

(ii) Voting → Assumption → each model should have min accuracy → 0.51, All model should be independent and dis-similar.

Regression, classification → multiple different model train on a dataset and most frequent (hard voting), Highest Probability (soft voting), mean will return.

(iv) Bagging → Boosting + Aggregation. Boosting (get random set of data from a data set with replacement), use only 1 type of Algo, at Prediction majority win and mean. • Types → Bagging, Pasting, Random subsequence, Random Patches, Bagging gives Better Result then Pasting, use Random Patchs and subspaces should be used while dealing with dimensional data.

(v) Random Forest → Difference Blw Bagging and Boosting →
① Bagging support all different algo, Random Forest only use decision Tree
② Bagging has model level column sampling (For each model there will be randomly columns get selected) where in random Forest has node level column sampling (For each node of every decision Tree, randomly columns get selected)

(vii) Bagging vs Boosting → Bagging → ① Types of models used:- models with low bias and high variance ② Parallel learning Possible ③ Base model weightage is equal. Boosting → ① Types of models used:- model with high bias and low variance ② sequential learning ③ Base model weightage is not equal

(viii) AdaBoost → stage wise additive method, Buid by adding multiple weak learns, all the weak learners are added stage By stage, weak learners= ml model who has very less accuracy (mostly we Decision Stumps as a group of weak learners (Decision Tree with max height = 1)) Note → In adaboost classification Problem → we use +1 and -1 instead of 1 and 0

(ix) Gradient Boosting → Sequential stage wise addition, GB uses the gradient (loss) of model as input to the next model and its goes on.

(X) Adaboost vs GB → ① Adaboost → we use Decision Stump which have max depth=1, we assign a different weight to each model

② GB → Here we use decision Trees which have max depth in b/w (8-32), we assign a single learning rate for each model.

(S12) Stacking and Blending Ensembles :- First we train multiple different models (called Base models) on data, then we give Base models output to a new model (called meta model) as input with data output columns There are 2 way to do this → ① Blending / Hold out method
② Stacking / K-fold-method

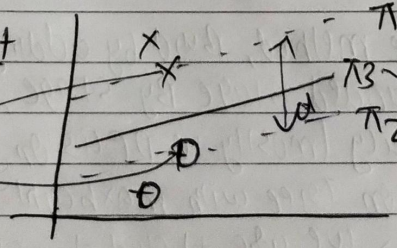
(S13) Hierarchical clustering → Type ① Agglomerative clustering ② Divisive clustering,
↳ mostly used, use dendrogram method to find the number of clusters.

(S14) K-Nearest Neighbors → KNN doesn't really "learn" from training data, it simply stores all the points and waits until it need to make a prediction (lazy learner), when a new data points comes for prediction KNN looks at K Nearest neighbors for prediction (majority w/n for classification and mean for regression) uses euclidean distances, Div → Slow for large data, outlier can effect, fail for high dimensional data. To decide the number of cluster make multiple model with Grid search CV and choose the best one.

(S15) SVM

core Intution = Maximize the margin b/w dataset, Base = logistic Regression logic

For a dataset these points called support vectors



① we choose that line which max margin (d)

margin maximize Plane

working → First we draw a line like π_3, parallel to which we draw 2 lines π_1, π_2 which is toching the support vectors, then we find out the d, max d line will be the answer

Benefits → ① Robust to outliers ② able to work for non-linear dataset with the help of Kernels.

Kernel Trick

→ used to classify data which is non-linear

→ Here we use a maths functions and applied it on data so that now that data set become separable.

→ it convert low Dimensional feature space into Higher Dimensional feature space