AI has ML has Deep Learning

ML HAS two types :

1. Supervised learning : data has features and 🡪 outcomes
2. Regression problem : continuous target, best fitting graph
3. Classification : targets discontinuous grouped into classes
4. Unsupervised : only have features, no outcomes
5. Clustering
6. Semi supervised : aka Q learning, write code so that something is done automatically, like chrome dino game, automated chrome dino, sudoku solver
7. Reinforcement learning

Data divided into 3 groups : training testing validation set

Exploring of data by data analyst :-

O : Obtain data

S : scrubbing or cleaning data

E : Exploring/ visualizing data to find patterns and trends

M : Model data for prediction, multiple analysis

N : interpret data, check accuracy, validate

Hello World of ML

<https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229>

one value mostly , dominating  Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.: bias

variance : result spread from actual

 There is a tradeoff between a model’s ability to minimize bias and variance.

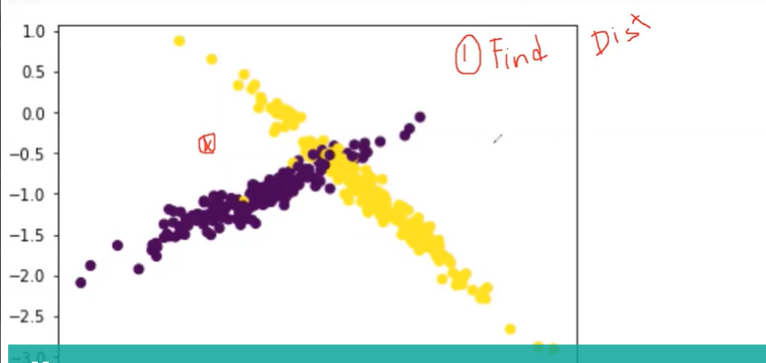
Some supervised learning algo examples

* K Nearest Neighbors
* Naïve Bayes
* Decision Trees
* Linear Regression
* Support Vector Machine (SVM)
* Neural Networks – can do regression classification both

Non supervised

* Clustering algos – K means clustering
* Associate rule learning algos

KNN (k nearest neighbors classification)



For any point x , we

1. Calculate distance from all the points

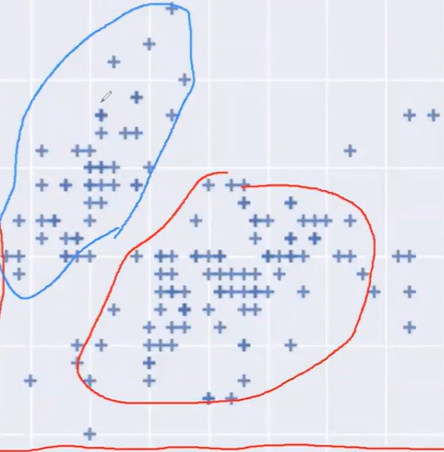
Gaussian distance is maths coordinate system dist sqrt( (x1-x2)2 + (y1-y2)2 )

Manhattan dist is ( abs(x1-x2) + abs (y1-y2) )

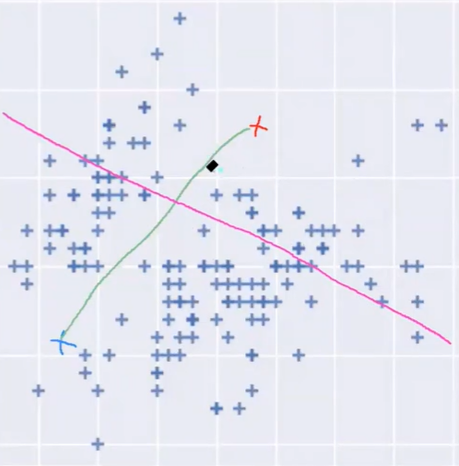
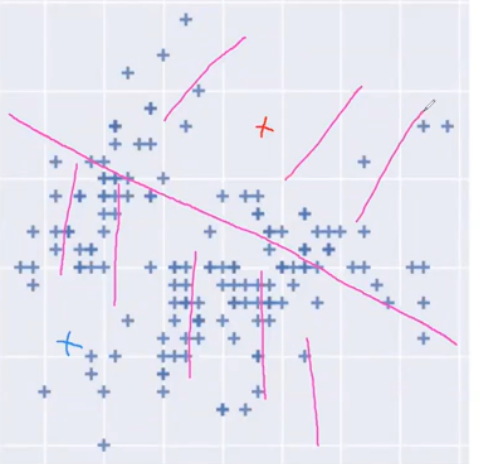
1. Find K distance that are close to this point, generally choose k as Odd but also prefer it as prime. So like 7 chosen, and 3 are yellow 4 are purple so it will be classified as purple

K Means Clustering

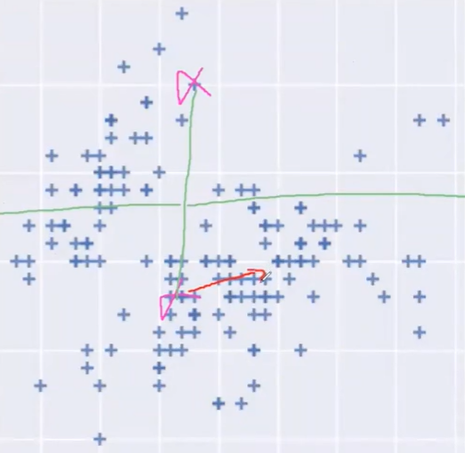
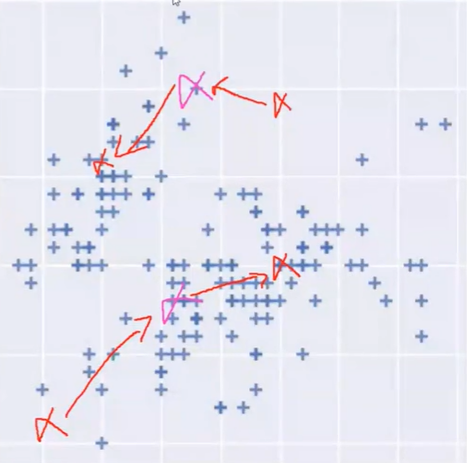
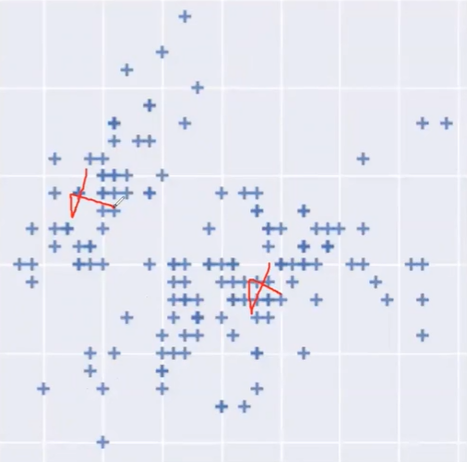
A un-supervised clustering algorithm so no target present, we make clusters/groups with this



Say we want to divide in two cluster so take two points randomly and draw its perpendicular bisector.

  this was one iteration

Now it moves the centers chosen to the centroid of the points or towards more swarm density in that cluster, again those which are closer to x1 are grouped with it and rest with x2.

  \

It can also be understood as initially it groups points randomly then calculates their centroid/mean, calc distances of points from those centers, whichever is closer to whichever point is grouped with it. Again iterate

After some time the centers stop moving, that can be considered as stopping point.

LINEAR REGRESSION

Regression with continuous target value present

Eqn of line : y = mx + c

The line which fits the graph best is considered.

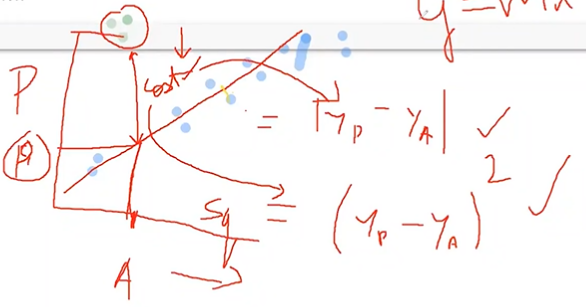
Say a relation between area and price. Then P = mA + c , we can get approx. price for this area. This predicted accuracy depends on fitting and data. Losses :-

Manhattan distance = | yp – ya |

Not very accurate results but gives realistic. 4 and 9 difference not as much visible tho it may be very large logically. This is not taken as its graph is not differentiable and not penalizing the values that are too far

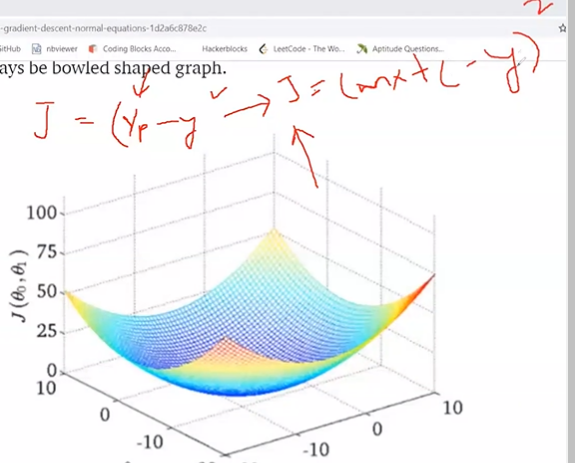
Square root distance = | yp – ya |2

Really high for points that are far, gives the results accurately. Accurate as difference between 4 and 9 very clearly told.



Loss or cost function = loss sum / no of observations

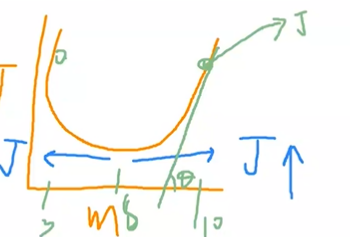
Sqr root loss will be a parabola/bell shaped curve c = ( yp – ya )2



We want to minimize our loss. To best fit. M and C are varying.

Cost J = I/M SUM(mx+c-ya)2

If c=0, the best fit line will have minimum J at say mi for various m (10degree, 30, 45, 60 ,75,90) in either directions the J will only increase, so J and m(slope) are related parabolically. Right side positive slope, left side negative slope. This is called “gradient decent” used to minimize cost



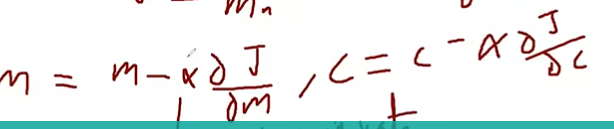
Yp = mx+c hypothesis

J = sum( Yp – Ya)2 / Mn

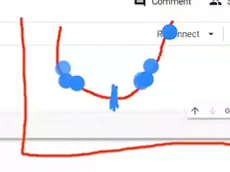
J = sum( mx+c – Ya)2 / Mn loss/cost function

To minimize cost function, using gradient decent

Hypothesis -> Cost to be minimized to predict the outcome



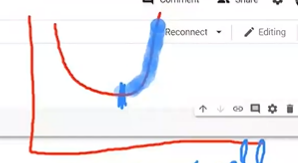
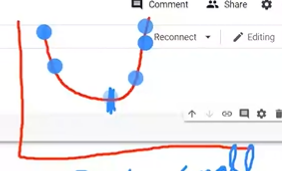
The decent is don’t at some interval or by some factor in every iteration, this is called alpha “the learning rate”. At first take big steps, then step size decreases and lastly becomes infitly small.

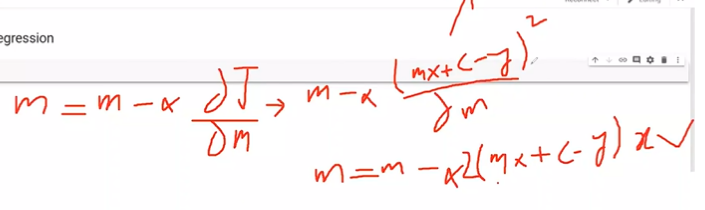


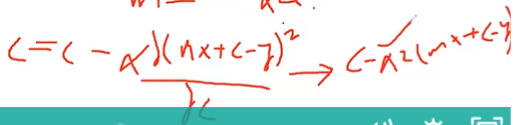
Case 1 : alpha is v small - Will take a lottt of time but will definitely reach small

Case 2 : alpha is big : big jumps so goes to / overshoots to the other side

So carefully need to choose alpha generally between [0.01 , 0.1]



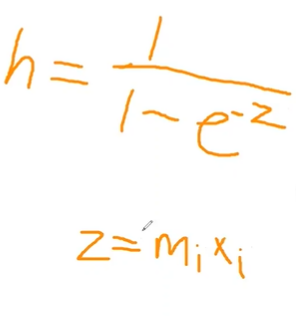


Classification Algorithms

LOGISTIC REGRESSION

Sigmoid function used instead of eqn of line

h = 1 / (1-e(-z)) , z = sum( mixi )

Penalty / Regularization : say we have y = m1x1 + m2x2 and x2 >> x1 so may overpower, thus, consider it as y = m1x1 + m2x2 – Lambda\*m2x2

Or y = m1x1 + m2x2(1-lambda)

if a parameter has very high value then penalize it by lambda

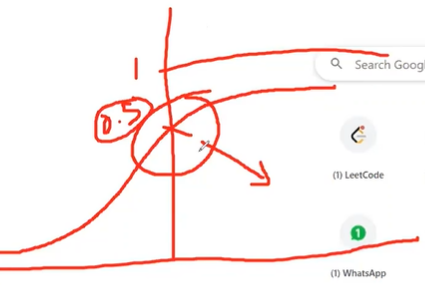
C : inverse of regularization

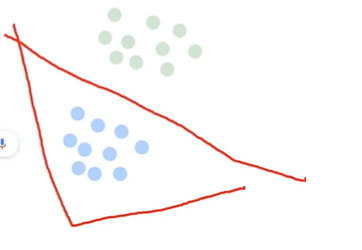
Solver = algorithm implementing for log reg like newton-cg, sag, ibfgs (default)

Multi-class or single class

SUPPORT VECTOR MACHINE

Log Reg has some problem in sigmoid function like at the intersection of region if a point comes, then which class to assign it as

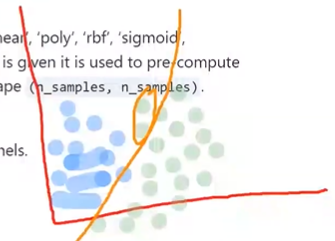
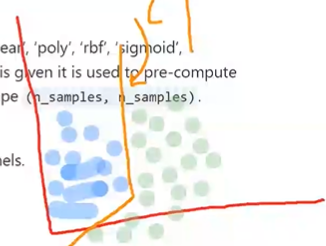


 log reg  SVM

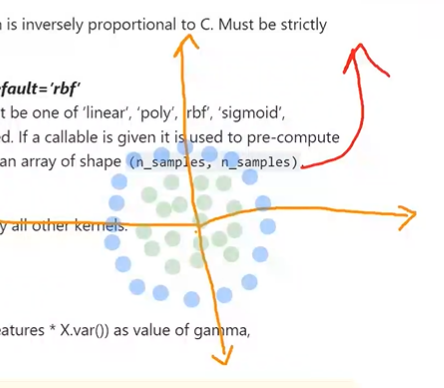
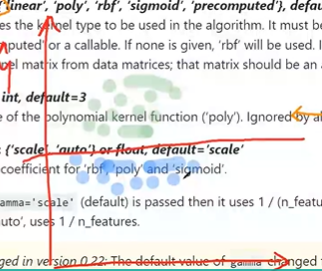
Svm also has 2 support vectors equidistance from the red line of Logreg. These help to differentiate between the classes and also that red line is doing it right.

Parameter :

C : regularization parameter. For low value of this it will simply draw classification line, even wrongly predicting, and high value may not draw straight line but will predict max high. So will keep it high 1000 or 100

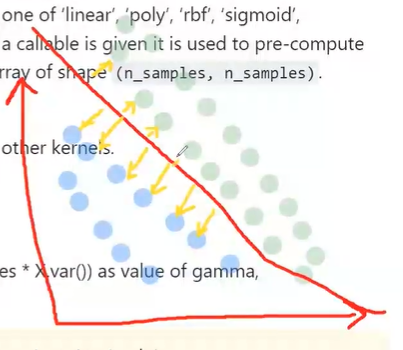
Kernel : no linear boundary to differentiate, so kernel changes all the points to another dimension to make a linear difference. Creates a diff plane.

Eg: rbf (gaussean kernel powerful) , poly (polynomial), sigmoid (log reg), linear(LinReg) kernels

Gamma : will keep it low. Kernel coeff for rbf poly and sigmoid.

If high gamma then will only consider points nearby to the line.

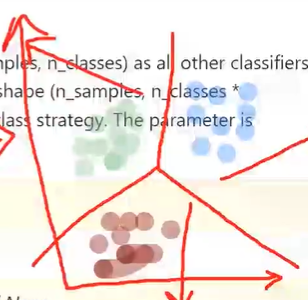
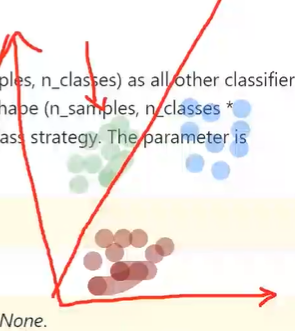
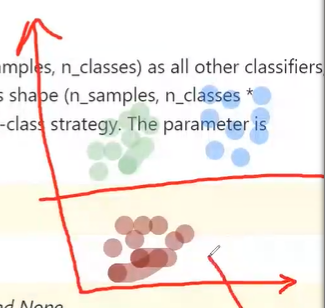


For low gamma, it will consider far points also so better line

Decision\_function\_shape : ovo (one vs one), ovr (one vs rest). Say need to predict between three classes.

Ovo creates a hyper boundry classifying all the classes

Ovr will consider one class at a time and all the other/rest classes as single class. Predicts with the help of all 3 lines.

DECISION TREES

Create like a Binary search Tree, classes are at leaf nodes, decisions at each inner and root node (like a<n in bst). Will use sklearn decision tree classifier, look at its documentation.

Parameters :-

Criterion : gini (default) , entropy. Function to measure the quality of split.

Entropy tries to minimize the entropy, so that randomness is reduced.

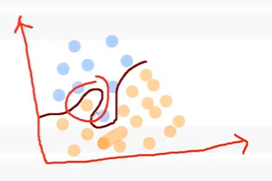
Gini we want to increase

https://blog.quantinsti.com/gini-index/

Max\_depth : to avoid overfitting use this max\_depth.

Decreasing max depth is called "Pruning"

Overfitting need to be avoided. Decision trees generally overfit



RANDOM FOREST

Combination of trees. Part of ensembeled learning.

Ensemble learning is the process by which multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular computational intelligence problem. Ensemble learning is primarily used to improve the (classification, prediction, function approximation, etc.)

Make many trees, see what they are predicting, ans is the most occurred one.

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Sklearn RandomForestClassifier parameters

N\_estimators

Criterion

Max\_depth

Min\_samples

Oob\_score

N\_jobs

Classifiers

* KNN
* Logistic Regression, SVM
* Decision Trees and Random Forests

Try using the last part of every point in order :

SVM <- RF, KNN <- LOG REG <- DT

Get data -> preprocessing -> visualize -> build models

ENSEMBLED LEARNING

Ask multiple people, make them choose between A and B ... N Say asked 20 people, choose the one which response has majority or most occurrence.



Train multiple classifiers, they can be of any type, like C1:SVM, C2:DT, C3:SVM, C4:LogReg etc. , they may be same type models but may have different parameters. They each produce results or outcomes based on the data they are given and the training they are given and the output of each is used for voting and most voted output is the considered final output.

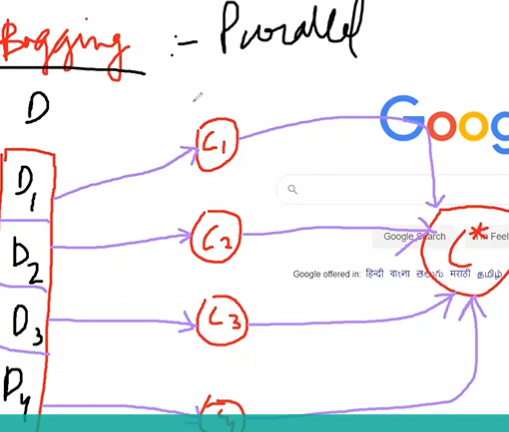
These C1, C2…Cn classifiers are called “weak classifiers”, while C\* is called “Strong Classifiers” having greater power, more precision and more accuracy.

Also Error is less of C\* than weak classifiers.

Has two types :

1. Bagging

Train weak classifiers parallelly. For n weak classifiers, divide data into n datasets and train each C on its set only, all parallelly.

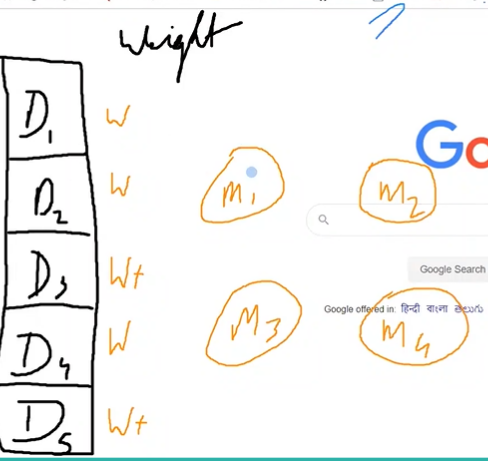
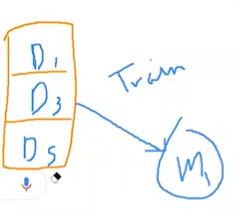


a. Random Forest : All classifiers are decision trees. Trees have a height of 1.

2. Boosting

Train weak classifiers sequentially. Say dataset has 5 subparts, each Di has an assigned weight to it Wi. Say we have models Mi.

Take a subset od dataset say D1,D2,D3 and then train M1 on it. Now provide full dataset to M1 for predictions, if it has wrongly classified D2 and D5 then increase weight of those wrongly classified (i.e. W2, W5).

Now select model M2, those rows with higher weights have higher probability of coming for training part of M2. Say D5, D3, D1 come. Train. Then ask M2 to predict on complete dataset. Say D5 got predicted wrongly so again W5++.

Select model M3, data subsets for training : D5,D3,D2. Train, Predict on all. Suppose this time correct predictions for all. So can stop. But not every time correct predictions will be generated.

With these Mi trained (weak classifiers), to use the model, we create a C\* (strong classifier) and give the input to all the Mi, again voting, and output the majority .



1. Ada Boost

<https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html>

n\_estimators=50 : no of weak classifiers

by default uses DecisionTrees as classifiers with depth=1. these are called "Stambed trees"

algorithm : SAMME, SAMME.R(default)

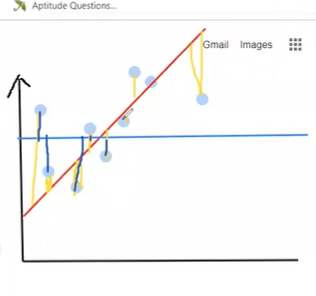
1. G boost
2. XG Boost ( very strong , most widely used)
3. Cat Boost
4. Light Gradient Boost

EVALUATIONS / SCORES

1. R2 Score : difference between predicted values and actual values.

= 1 – sum(Yp – Ya)2 / sum (Ya – Ym)2  . Lies between 0-1 and higher the value better the mode.

Red is Regression line, Blue line Mean of all points, yellow distances are Numerator, dark blue distances are Denominator.



1. Mean Squared Error (MSE)

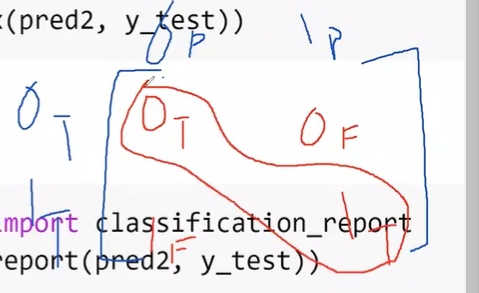
= sum(Yp – Ya)2 / N

RMSE : underroot of MSE

1. Mean Absolute Error (MAE)

= sum | Yp – Ya | / N

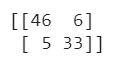
1. Confusion Matrix



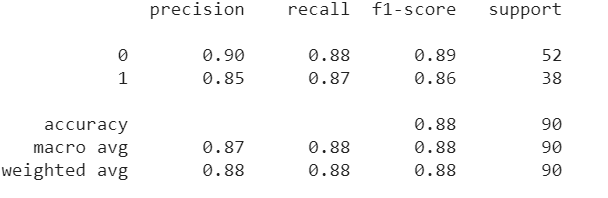
Rows : Actual 0, Actual 1… Columns : 0 predicted, 1 predicted

So cells : 0 correct prediction 1 Wrong predictions

0 Wrong predictions 1 correct prediction



1. Classification Report



Support : tells about no of rows that were having that class. So 0 class was in 52 rows, and 1 in 38 rows

Precision : 1T / (1T + 0F). Correct prediction of 1 per total predictions made as 1. Done from values of a column of the conf matrix.

46/(46+5) = 0.9019 33 / (33+6) = 0.846

Macro Avg : sum(precision) / N

(0.90 + 0.85 ) / 2 = 0.875

Weighted average : 0 is given certain W0, and 1 also has certain weight. It is the average of it.

Recall : 1T / 1total 1 predicted correctly / 1 total

46 / 52 = 0.88 33/38 = 0.86

F1 score : v imp, combination of precision and recall

= (2\*P\*R) / (P+R)

(2\*0.90\*0.88) / (0.90+0.88) , (2\*0.85\*0.87) / (0.85+0.87)

(0.8898876404494382, 0.8598837209302325)

Regressive problems : R2 , MSE , MAE

Classification : Conf. Matrix, Classification Report

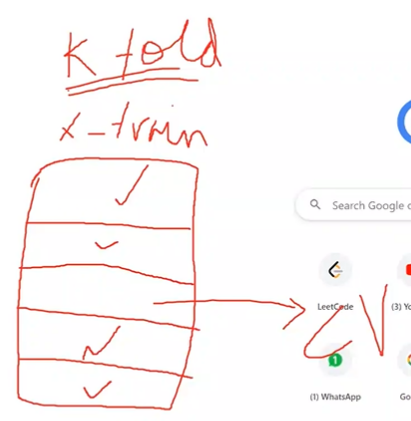
K fold - Cross Validation

Divide xtrain in say 5 parts.

Iteration 1 : select any 4 for training, cross validate on 5th subset.

Similarly 5 iterations with different 4 – 1 subsets from xtrain.

At last find the mean of all the scores.



Grid search : Hyper parameter tuning ( select best vals for the parameters