

K Macline Learning:

* Simple Linear Regressions

y= mzetb

- inewitable margin of error colled loss.

- Difference between predicted y &

actual y is called residuar.

- minimize residually using a loss function

Best approach is to square & sum them

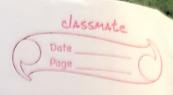
we can find m& b values that will find

the sum of least squares.

· solving for a fit

we rondomly increase/decrease mab with random values from a standard normal distribution or even better

a T- distribution hich has feeter teils



Note: Diffrient ways to perform linear regressings

(5 ways are given in books)

* multiple linear regression

solving for function ys meth's elementary as its only los one independent variable?

We can also solve for multiple Independent variables like 24, 22, 22, and so cn. --.

y = Po + B, 2, + B22e2.

the prof multiple.

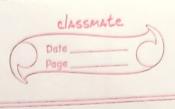
we marked processe in the

want distribution in considerate

ist producted and and remideral it

Suchmit pull sailer mobier die

we can find in the values of



Hill climbing - A simple optimization Algerithm.

1. Start with random or initial
Solution, even if its poor quality
2. Repeat the Collanding Steps for

2. Repeat the following steps for a number of iterations and/or until the solution cannot improve anymore

- Select a random part of the Solution - If that results in improvement,

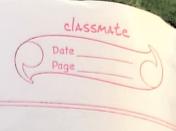
* K- means chatering

Take culorage of Ly value. By palue. By palue.

eres track of biordies done to that ans

at the gravage of their paints

Keep H.



* clustering

Let's start with A Centroids.

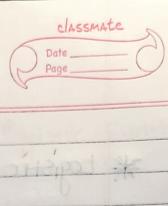
Distance between a point ¢roid.

- If that result in improve

d=1(1)2

* K- means clustering

- 1) Take average se & y values for points neavest to each ear Controid. (hence k-means).
- 2) Set each controid to that average 22 24
- 3) Repeat until the centroids do no not move anymore & gre are the average of their points.



Overfitting means that our me model works great with training data but fails to predict correctly with new dates.

High Variance & low bias.

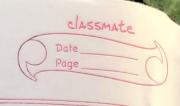
Splitting the duter into two sets training & testing with more data as

we can also train with more date as well as afflize cross-validation, regularization, bagging , boosting 4

regularization, bagging, boosting & other techniques.

: Os wildedolog tollong 9

100 Mon 40.1) 10.100



* Logistic Regression.

Logistic regression is a classification tool tenat predicts true or false value for one or more variables.

- An S shaped Curve (logistic Curve) is fit to the points & then used to predict probability

- It predicted value is less than

O.s it is typically categorized as

talse (a) & it the predicted value

is greater than | equal to 0.5 it is

typically categorized cas true (1)

J= 1.0+ e-(B0+B/2)

def predict_probability(20):

P=1.0/(1.0+ math.exp(-(bo+b)2e)) return p



you may also see this logistic function.

Notice has the expression BotB, se is linear, I this is known as long adds function. which is translated logarithamically into a probability

- · maximum Likelinood.
 - Cret true (1) duter, carlculate the likelihood y for each se value, and multiplied them toghether.
- (get the failse (o) date, calculate likelihood (1.0-y) for each se value & multiply them toghether.
- multiply the two products above teghether, & that is your total likelihood.



to avoid floating point &

we can use logarithmic addition instead of multiplication

So modified function is

- pass the value through log 174 add.
- pass the value through leger &
- Sum the two values above toghthe pass & it through exp() function to undo loganithm & that is total likelihood.

$$y = \log \left(\frac{1.0}{1.0 + e^{-C-3.1V + 0.692e}} \right)$$



* Naive Bayes is a machine learning application of bayes theorem that merges probabilities of multiple

features to predict a given Category.

Pfe = Probability of featurese; with a fudged constant in

Occur product = exp(leg(pf,)+log(pfs)

+ leg(pfn))

numerator & denominator

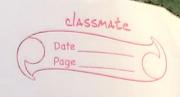
Not ochur Product = exp (log (1-Pg)

+ log (1-Pg) + --+

log (1-Pg))

Combined probability = (occur product)?

(occur product) + (Not occur product)

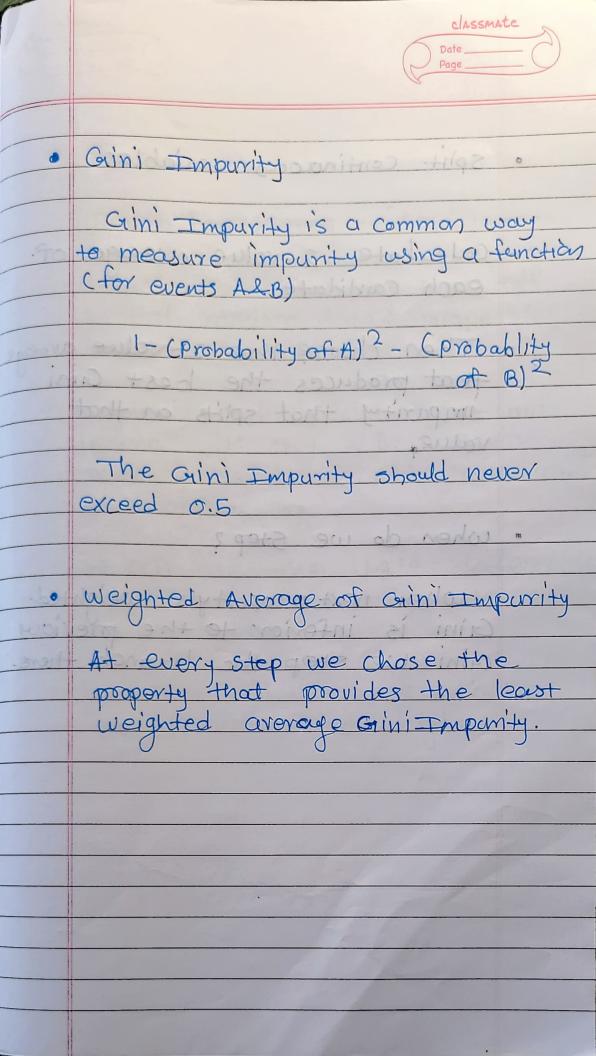


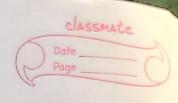
* Decision Trees.

Decision trees are a powerful machine learning tool & work well for a lot of machine learning problems.

- Decision trees are notorious of
- This can be remedid by Random Forest.
- Decision trees can also be improved with gradient boosting
- Other flavours of pecision trees exists like regression trees.

note: - x GBoost is considered best in prediction type of problems.





- · split continuous variables.
- Calculate 2-value average of. each candidate
 - Then choose the 2-value average that produces the best Gini impurity that splits on that value,
- · when do we stop?

Crini is inferior to the previous Crini we stop the branch there.



* Random Forest

Random Forests are machine learning that generates hundred of decision trees, whore each one build off partial random data & praperties, yather than all data

- Typically each decision trees will learn with only 2/3 of the randomly sampled dota, which is known as loootstrapping
- Hown as out-of-bag deuter as
 the test double.
- Each tree will vote on prediction, the prediction with highest vote wins.