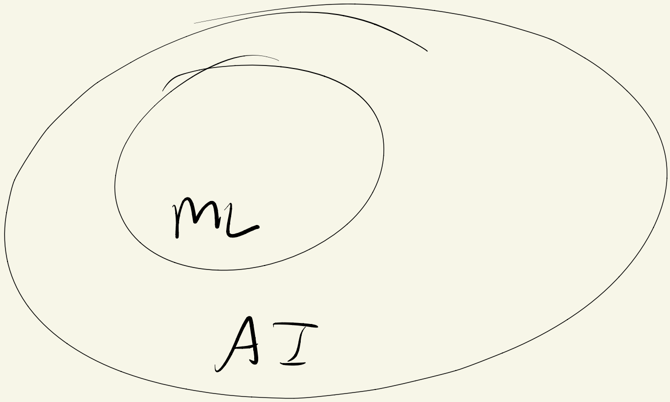
Week 1



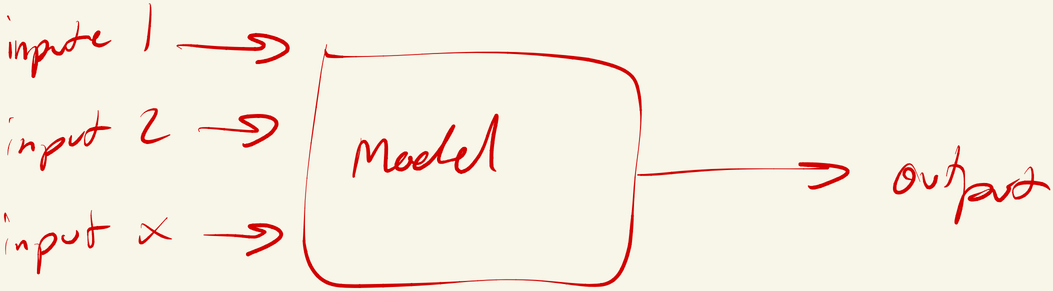
DS - Data Science

1. Supervised Learning - using labels to train models & learn outputs
   label cat picture, label dog pictur

2. Unsupervised Learning - unlabeled data to learn patterns between multiple pictures

3. Reinforcement learning - reward / penalty for interactive environment to learn

3 types of machine learning

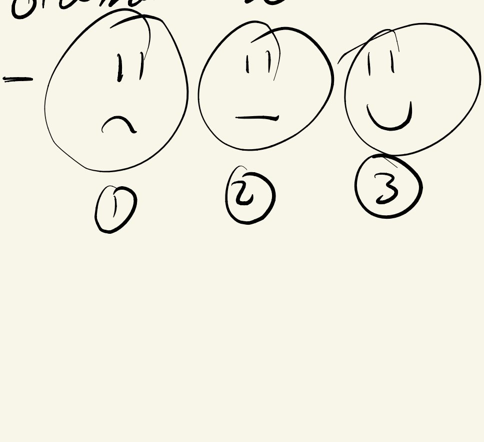


- Qualatative data have finite number of groups (w/ no inherant order)

  nominal data

  - one-hot encoding
    - if matches category = 1
    - if not = 0

- ordinal data — have inherant order



Ⓠ Quantantive Data - numerical
discrete or continous

supervised

① Classification - predict discrete class

- multiclass classification
  ex. hot dog, pizza, ice cream
  ex. fruits, animal, plant species
- Binary classification
  ex. hot dog, not hot dog
  ex. cat/dog, positive/neg, spam/not

② Regression - predicting continous values
  ex. price of ETH, temperature

Week 2 : Training model

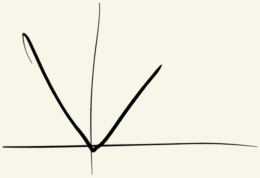
- feature vectors all all at
  petlents quantative descriptions
  ↳ all ver on one rule (features measuring to use as input)
- feature matrix: all data
- target vector → prediction
  aspect – "output" – the dependent variables

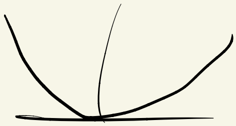
How well can this model handle new data

- Training dataset → model → prediction vector
- Validation dataset
  - minimize lost as much as possible
  ↳ find beest model
- Loss: actual diffence between reality & prediction
-

Loss functions

L_1 $|y_2 - y_1|$ $\qquad$ L_2 $(y_1 - y_2)^2$





- Binary Cross-Entropy Loss

Performance ↑ → ↓ loss

- Scale #'s to be scaling

---

output = target

- Train model on only part of data
- Model can be "overfitting" - can predict within that data set do not well, be good for data model but not new data
- Validation Set (2^{nd}) - reality check to ensure it can handle new data
  - > Loss is for user to see best model
  - > validation data used to test models against each other
- Loss is putting models against eachother

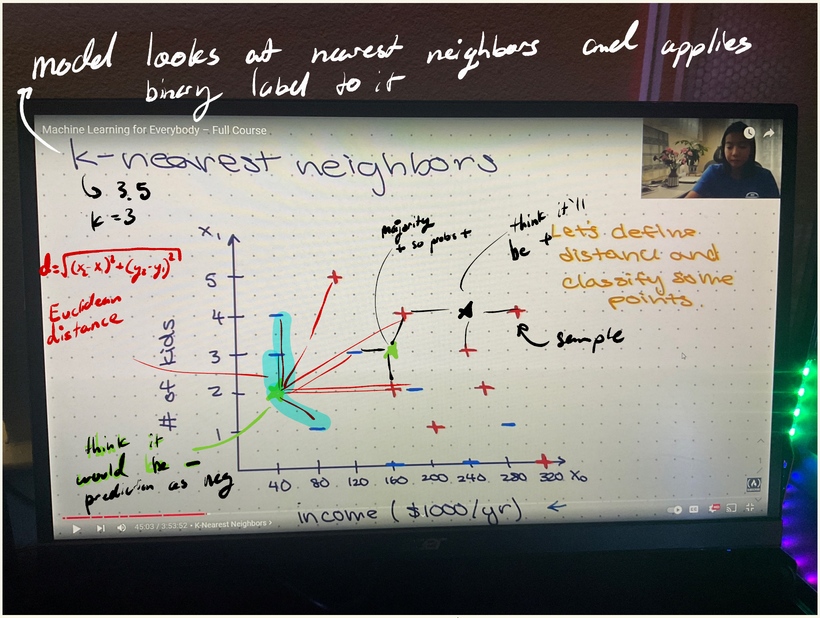
Notes 2/4/23

K-nearest neighbors

Binary

+ = own car

- = no car

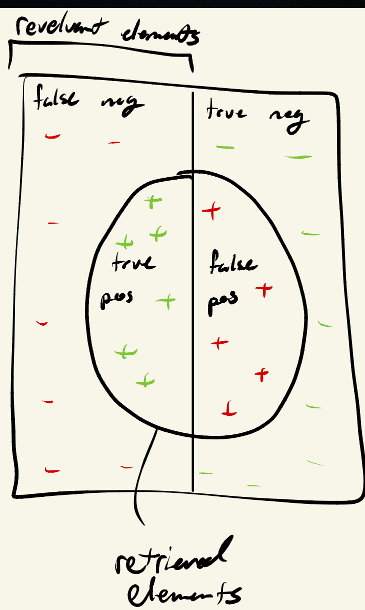


How many true positive

Precision : $\frac{+}{+ \,\&\, +}$ aka $\frac{\text{true positive}}{\text{retrieved elements}}$

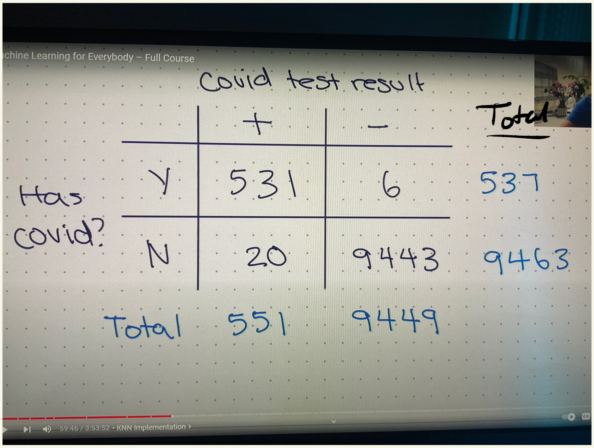
Recall : $\frac{+}{+ \,\&\, -}$ aka $\frac{\text{true positive}}{\text{relevant elements}}$

Out of all we labeled true positive, how many are truly positive



- Odd value of k eliminates ties
- not binary classification → 3 categories
- tie → which occurs first in data points
- 10 dimensions in our model
- more data points → more noise → can choose a value of k too high / locality

$$P(\text{covid} \mid + \text{ test}) = \frac{531}{551} = 96.4\%$$



Bayes Rule

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Naive Bayes

– assuming no correlation between any variables

$$\underbrace{P(C_k \mid x)}_{\text{posterior}} = \frac{\overbrace{P(x \mid C_k)}^{\text{likelihood}} \cdot \overbrace{P(C_k)}^{\text{prior}}}{\underbrace{P(x)}_{\text{evidence}}}$$

x: feature vec

– ML is pointing us in right direction

$$P(C_k \mid x_1, x_2, \ldots, x_n) \propto P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k)$$

Π for product of multiple things

$\hat{y}$ is predicted y

$\underbrace{P(C_k \mid x_1, x_2, \ldots, x_n)}$: prob that we in C_k given we in $x_1, x_2 \ldots x_n$

ex. given that rainy, not windy, and it's Wednesday

$\propto$ means proportional to

$$P(C_k \mid x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n \mid C_k) \cdot P(C_k)}{P(x_1, x_2, \ldots, x_n)}$$

$P(x_1, x_2, \ldots, x_n)$ → constant for all classes so it's proportional

$$P(C_k \mid x_1, x_2, \ldots, x_n) \propto P(x_1, x_2, \ldots, x_n \mid C_k) \cdot P(C_k)$$

$$P(C_k \mid x_1, x_2, \ldots, x_n) \propto P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k)$$

$\hat{y}$ is predicted y:

$$\hat{y} = \underset{k \in \{1, K\}}{\operatorname{argmax}} P(C_k \mid x_1, x_2, \ldots, x_n)$$

$k \in \{1, K\}$ – k class that maximizes probability

MAP: Maximum A Posteriori — can apply for any # of categories – only 2 in our case here

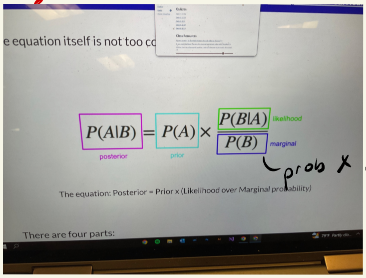
aka pick most probable k to ↓ probability of misclassification

– Ignoring depencies bc point of algorithm isn't to for determing, it's % prob between the two

– Gaussian → normal

– Naive bayes → map

↳ everythings indep

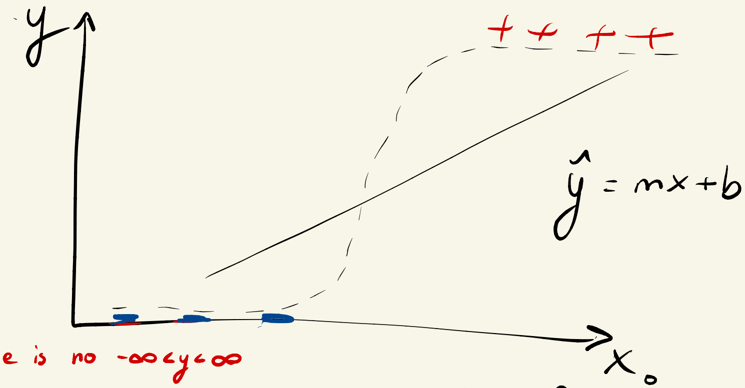


$P(A \mid B) = P(A) \times \frac{P(B \mid A)}{P(B)}$

$P(B)$: ↳ prob X of occuring

Notes 4/17

Logistic Regression



range is no $-\infty < y < \infty$

ratio can take on infinite values

How can we use this model for probability?

$$e^{\ln\left(\frac{p}{1-p}\right)} = e^{mx + b}$$

$$\frac{p}{1-p} = e^{(mx+b)}$$

$$p = e^{mx+b} - pe^{mx+b}$$

$$p + pe^{mx+b} = e^{mx+b}$$

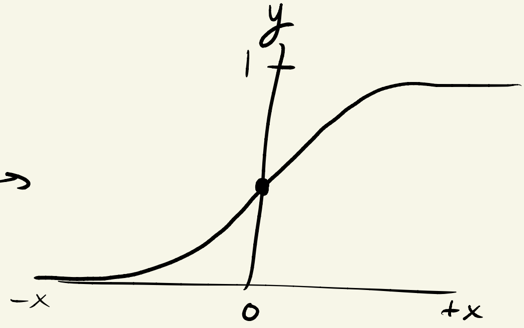
$$p(1 + e^{mx+b}) = e^{mx+b}$$

$$p = \frac{e^{mx+b}}{1+e^{mx+b}} \cdot \frac{e^{-(mx+b)}}{e^{-(mx+b)}}$$

$$p = \frac{1}{1+e^{-mx+b}}$$

Sigmoid Function

$$S(y) = \frac{1}{1+e^{-y}} \longrightarrow$$



Sigmoid function

$x_0 \rightarrow$ simple log reg

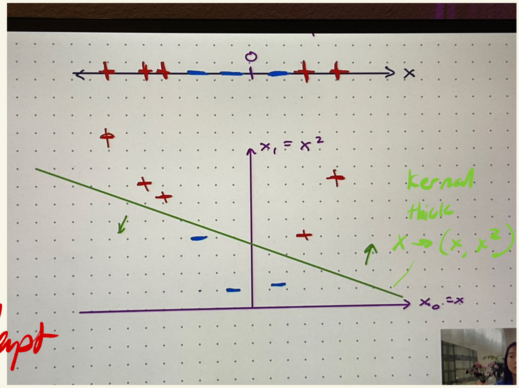
$x_0, x_1, x_2 \rightarrow$ multiple log reg

- logistic regression: dichotomous & boolean variables
- multi-nomial logistic regression: treating every category true/false
  - can handle more than 2 in target variables
- simple logistic vs multiple logistic
  1 feature independ — multiple features in indep variables

3 tests so far
- K nearest neighbors
- Naive Bayes
- Logistic Regression

Support Vector Machines (SVM) - 39:43

- line / plane that devides points — above 3 → hyper plane → distance between
- Margin : the points distance from the line on both sides of line.
  - > support vectors → define boundaries are the margins from the line



- outliers sincerely hurt SVM
- 87% accuracy on SVM
- goal is to maxize margin
- weakness → outliers effect line
- not robust to outliers, need to adapt
- outliers definition ~~basic~~
  - ↳ 7 data keep
  - ↳ 5k, maybe remove
  - ↳ outliers are where interesting thing happens
- kernell trick → transformation to data
- stastical classification wiki

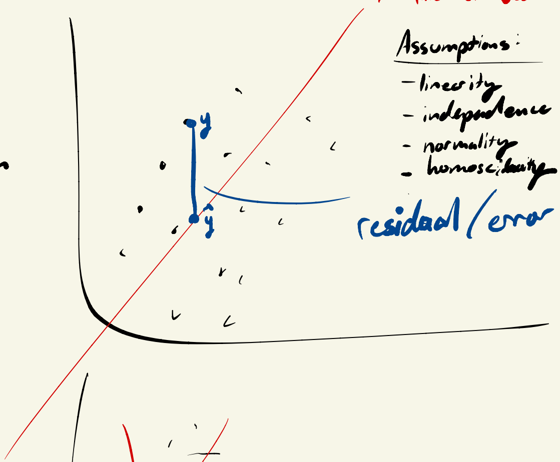
Lineer Regression

<u>Simple</u>: $y = b_0 + b_1 x$

<u>multiple</u>: $y = b_0 + b_1 x + b_2 x_2 + b_n x_n$

simple linear regression

line of best fit



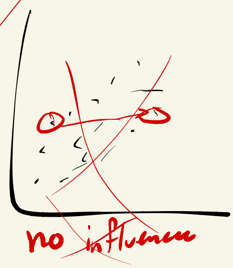
Assumptions:

- linearity
- independence
- normality
- homoscidasity

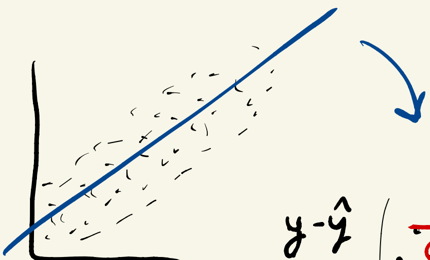
residual / error = $|y - \hat{y}|$

Assumption:

linearity



independent

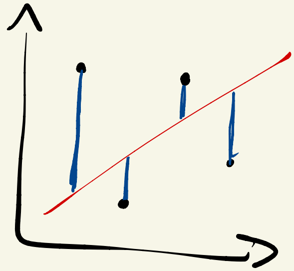


normality & homoskedasticity

these two sections are equal

other not 

Evaluating Lin Reg Model



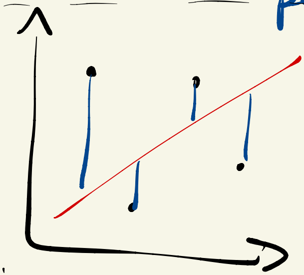
① Mean abs error (MAE)

$$\frac{\sum |y-\hat{y}|}{n}$$

- can compare across units

let n = number of points

② Mean squared error (MSE)



$$MSE = \frac{\sum (y-\hat{y})^2}{n}$$

- punishes for larger errors
- can't directly compare values w/ y-axis

③ Root mean square error

$$RMSE = \sqrt{\frac{\sum (y-\hat{y})^2}{n}}$$

④ R^2 - coefficient of determination

$$\boxed{R^2 = 1 - \frac{RSS}{TSS}}$$

let RSS: sum of squared residual

RSS: $\sum (y-\hat{y})^2$ — punishing larger errors

let TSS: total of sum of squares

$$TSS = \sum (y-\bar{y})^2$$

Random Forest Classifier