بسم الله الرحمن الرحيم

أول ماشين ليرننج تكنيك .. وهو ال Nearest neighbor ... مراجعه سريعه علي الي خدناه المره اللي فاتت ... عندك شوية إنبوت X و عندك function f احنا مش عارفينها .. والسؤال هل احنا نقدر نشوف فانشكن f تقدر تماب الانبوت للاوتبوت .. فاحنا بنسيرش علي hypothesis h عشان نقدر نحدد ال f ... سؤال سريع ... ده اسمو inductive learning ... ايه الفرق ما بينو و ما بين ال deductive learning .. محدش عارف الاجابه .. الديداكتيف هو ان عندك شوية رووزلز وشوية كونسبتس .. و بيبقا عندك شوية أمثله وبعدين يقولك حط الرووزلز علي الأمثله و ت deduce ايه اللي يتعمل في الامثله اللي عندك .. الإنداكتيف ليرننج هو العكس .. انت عندك شوية أمثله و انت عاوز تعرف ايه هي ال underlying functions اللي بت exaplin الامثله ديه ...

لما هنتكلم علي ال supervised learning في 2نوعين من المشاكل .. اول حاجه هو الكلاسيفيكيشن و تاني حاجه هو ال regression ..

## Supervised Learning

- Two types of problems
  1. **Classification**
  2. **Regression**
- NB: The nature (categorical or continuous) of the domain (input space) of $f$ does not matter



في الكلاسيفيكشن: ال range (output space) of function f is categorical

في الريجريشن: ال range (Output space) of function f is continuous

ركز ان النوع هنا بيفرق علي الاوتبوت .. مالوش دعوه بالانبوت ..

تعال نبص علي أمثله .. هنبص علي setting .. عندك outdoor sport .. وانت عندك implicit function .. وبناءً علي الجو هتقول إنك يا تستمتع بيه او مش هتستمتع بيه ..

## Classification Example

- Problem: Will you enjoy an outdoor sport based on the weather?
- Training set:

| Sky | Humidity | Wind | Water | Forecast | EnjoySport |
|---|---|---|---|---|---|
| Sunny | Normal | Strong | Warm | Same | yes |
| Sunny | High | Strong | Warm | Same | yes |
| Sunny | High | Strong | Warm | Change | no |
| Sunny | High | Strong | Cool | Change | yes |

$x$ $f(x)$

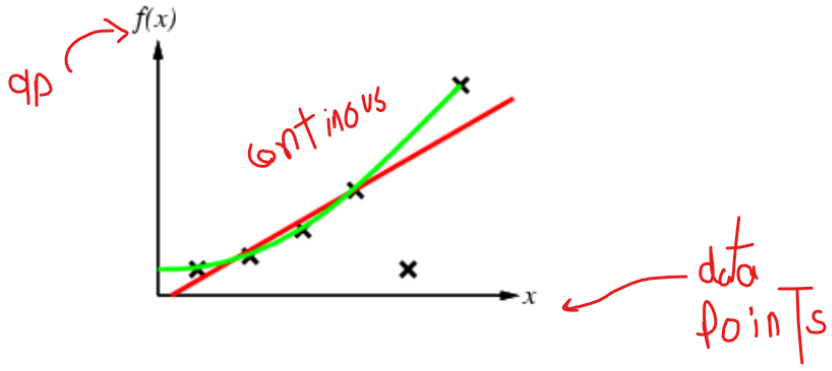
- Possible Hypotheses:
  - $h_1: S = sunny \rightarrow enjoySport = yes$
  - $h_2: Wa = cool$ or $F = same \rightarrow enjoySport = yes$



الفانكشن f هي مش عندك .. انت بتعملها induce ..

# Regression Example

- Find function $h$ that fits $f$ at instances $x$






تعال ناخد شوية أمثله من علي أرض الواقع ..

**اول مثال عندنا هو ال spam detection ..** لادكتور سال سؤال .. ايه هو الانبوت بتاع الspam detector .. حد قالو اغلب الانبوت بيقا تكست .. سال تاني ازاي نقدر نربرزنت التكست .. حد قالو هنعمل فيكتور و عندنا أكتر من representation .. حاجه مثلا زي one hot vector او vector .. embedding

فال Domain هيبقا text ... اللي هو vector of words

ال range هيبقا{spam, legit}

**تاني مثال عندنا هو ال Sock prediction**

ايه هو الانبوت: Time series of prices

ايه هو الاوتبوت space: {up, down} or R+ (all positive numbers)

خد بالك هو لو regression problem انت هيبقا عندك ارقام مش بس ال Real .. بس في اغلب الوقت ممكن تعمل كدا ..

**تالت مثال هو ال speech recognition ..**

ايه هو الانبوت: Voice (time series of audio signal)

ايه هو الاوتبوت: هي ممكن تبقا هو مين اللي بيتكلم ..او انك تبقا مهتم بإيه هو اللي بيتقال .. لو انت خدت في الاعتبار الكلمات ... هيبقا ايه هو نوع المشكله

ايه هو النوع: Classification عشان الحاجات اللي بتقولها هي discrete ... بس لو حد كان اتعرّض لل NLP .. الكلمات بتبقا encoded using some embedding ... هل كدا بقت regression ولا لا؟ .... لا .. لإن في الآخر هي حاجه واحده اللي هتطلع مش حاجات كتير ورا بعضها ..

**رابع مثال هو ال digit recognition:**

الانبوت في الحاله ديه هي images

الاوتبوت هو الارقام بتاعتنا

والنوع هو classification

**خامس مثال housing evaluation:**

الانبوت هو معظم ال features بتاعت البيوت يعني .. House features

الاوتبوت هو ال positive real number R+

ديه هتبقا regression problem

**آخرر واحد هيبقا ال Weather prediction**

الإنبوت هو sensor data من ضمنها ال satellite imagery ..

الاوتبوت هو درجة الحراره الي هي رقم حقيقي و {Rain, sun}

نوعها : ريجريشن لدرجة الحراره و classification للشمس و المطر

يلا نكمل ... الدكتور بيقول ان احنا عشان ندور علي ال Hypothesis هنحتاج ندور في space .. اسمو ال hypothesis space

## Hypothesis Space

- Hypothesis space $H$
  - Set of all hypotheses $h$ that the learner may consider
  - Learning is a search through hypothesis space
- Objective: find $h$ that minimizes
  - Misclassification
  - Or more generally some error function

  with respect to the training examples
- But what about unseen examples?

University of Waterloo CS480/680 Spring 2019 Pascal Poupart 7

في ال space ده انت ممكن تدور علي optimization .. وهو ده ال objective .. احنا لما بن mimize h .. في ال unseen هم جايين من نفس ال distribution .. الهدف في الآخر تلاقي hypothesis تقدر ت generalize ..

## Generalization

- A good hypothesis will generalize well
  - i.e., predict unseen examples correctly
- Usually …
  - Any hypothesis $h$ found to approximate the target function $f$ well over a **sufficiently large set of training examples** will also approximate the target function well over any unobserved examples

University of Waterloo CS480/680 Spring 2019 Pascal Poupart 8

في حاجه مهمه .. لما تلاقي hypothesis .. اتاكد انها تتوافق مع كل ال examples .. ديه حاجه انت عاوزها .. انت عاوز تلاقي hypothesis ماسكه نفسها معظم الوقت .. بس مش دايماً بتبقا متاحه .. ده هتلاقيه فعلياً مش متاح في حالات اللي بيبقا فيها insufficient hypothesis space .. فمثلا يعني لو عندي function f والمعادله بتاعتها زي منت شايف في السلايد .. ديه فانكشن انت بتحاول تكتشف هي ايه .. و حالياً قاعد ياعيني بتدور علي الفانكشن في ال space .. أين أنتي يا فانكشن .. بس اللي انت مش واخد بالك منو هو ان السبيس اللي بتدور فيه هو space of polynomials of finite degree .. فانت مش هتعرف تلاقي اللي انت عاوزو في ال space ده .. الدكتور سأل سؤال .. حد يعرف ليه الفانكشن اللي هي ax + b + xsin(x) ليه مش هتلاقيها في ال space of polynomials of finite degrees؟

المشكله في ال sin function .. لو عملتلها representation by taylor series of expansion .. هتلاقي ان عندك series طويله .. بس ال degrees بتاعت ال polynomials اللي هتطلع من ال expansion هتكبر و هتعوز منك infite large degree for the polynomial

فانت مش هتقدر ت represent sin(x) function ..

تاني مشكله هتظهر هو ال Noisy data ... مش هتلاقي علي طول كدا general consistent hypothesis

# Inductive Learning

- Goal: find an $h$ that agrees with $f$ on training set
  - $h$ is **consistent** if it agrees with $f$ on all examples
- Finding a consistent hypothesis is not always possible
  - Insufficient hypothesis space:
    - E.g., it is not possible to learn exactly $f(x) = ax + b + xsin(x)$ when $H$ = space of polynomials of finite degree
  - Noisy data
    - E.g., in weather prediction, identical conditions may lead to rainy and sunny days



لو انت مش قادر تلاقي general consistent hypothesis .. غالباً هتقول إن المشكله unrealizable .. و الكلمه unrealizable .. هي مش وحشه يعني .. في ال practice .. هي بتقولك بس تعال نلاقي hypothesis space كبير جداً .. عشان نقدر نلاقي as many functions as possible .. واحنا في الآخر فعلاً بنزود احتمالية اننا نلاقي ال function اللي عاوزنها .. في trade off ما بين expressiveness و complexite .. لو عندك large hypothesis space .. هو expressive أكتر وممكن فعلاً تلاقي hypothesis كويسه جدا.. بس انت محتاج تدور .. و التدوير بياخد وقت و data كتيره .. وبالتالي فلوس كتيره .. فهتحتاج داتا كتيره عشان انت هتعوز تقارن hypothesis كتيره .. فمش دايماً شرط اني أخش علي طول علي largest hypothesis space .. بسبب ال tradeoff ده .. احنا بنحتاج نبقا عندنا space of reasonable functions .. وتشتغل

# Inductive Learning

- A learning problem is **realizable** if the hypothesis space contains the true function otherwise it is **unrealizable**.
  - Difficult to determine whether a learning problem is realizable since the true function is not known
- It is possible to use a very large hypothesis space
  - For example: H = class of all Turing machines
- But there is a **tradeoff** between **expressiveness** of a hypothesis class and the **complexity** of finding a good hypothesis



دلوقت احنا جاهزين نتكلم علي أول machine learning technique

أول حاجه هنعرّف ال nearest neighbor classifier ب الفانكشن اللي اسمها h(x) .. ديه بتاخد انبوت query(x) .. و هتقوم مرجعه label y .. وده هيبقا ال label بتاع أقرب نقطه *x .. فهنا *x هي نقطه ليها أصغر مسافه .. والمسافه بنقيسها بطريقه معينه زي اللي في السلايد كدا ..

# Nearest Neighbour Classification

- Classification function

$$h(x) = y_{x^*}$$

where $y_{x^*}$ is the label associated with the nearest neighbour

$$x^* = argmin_{x'}\, d(x, x')$$

- Distance measures: $d(x, x')$

$L_1$: $d(x, x') = \sum_j^M |x_j - x'_j|$

$L_2$: $d(x, x') = \left(\sum_j^M |x_j - x'_j|^2\right)^{1/2}$

...

$L_p$: $d(x, x') = \left(\sum_j^M |x_j - x'_j|^p\right)^{1/p}$

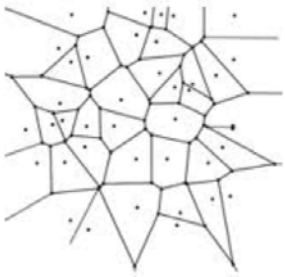
Weighted dimensions: $d(x, x') = \left(\sum_j^M c_j |x_j - x'_j|^p\right)^{1/p}$



في حاجه اسمها L1 و L2 .. L2 هي ال Euclidian distance .. سؤال حلو ... ازاي نفهم ايه اللي بيحصل في حالة ال classification .. انت بتقسم ال space .. انت عندك داتا بوينتس .. كل نقطه هي داتا بوينت .. و انا بشغّل ال nearest neighbor classifier . انت تقدر تتخيل ان في شوية regions عندك .. جواهم شوية نقط و أي نقطه انت عاوز ت query هت map to the closest point . وبعدين ترجّع ال Label بتاع النقطه ديه .. هو بيقسم ال input space لشوية مناطق . وال label بيبقا لكل منطقه .. ال partitioning معروف بإسم Voronoi diagram .. وده بنجيبه عن طريق انك بتعمل nearest neighbor باستخدام ال nearest neighbor.. .. المناطق هتظهر عن طريق ان عندك boundaries علي هيئة straight lines .. هتبقا في النص علي طول ما بين ال 2 ريجنز ...

# Voronoi Diagram

- Partition implied by nearest neighbor fn $h$
  - Assuming Euclidean distance





لما بتيجي تشتغل مع ال nearest neighbor .... هو unstable .. ليه .. لإن الفكره مبنيه علي اننك ترجع ال class بتاع النقطه اللي اقرب حاجه لل query بتاعك .. بس الفكره هنا ايه اللي هيحصل لو انت عندك شوية noise .. في الداتا اللي عندك .. الداتا بتيجي من شوية سينسورز ... والسنسورز بيبقا ليها إيرور في القرايات .. فبالتالي اللي انت بتفكّره أُقرب حاجه ليك .. هو مش أقرب حاجه ليك .. او ممكن يكون فعلاً اقرب حاجه ليك .. بس هو متسمّي غلط .. فالسؤال الجامد جداً هو ازاي تخلي ال nearest neighbor يبقا more robust to noise .. أول فكره هو إنك ت consider K-Nearest neighbor .. بدل ما انت بت produce a label based on a single nearest neighbor .. خلينا ن consider شوية neighbors و نبص علي most frequent class for that set of neighbors وبعدين هب نرجعهم .. فبكدا حتي لو نقطه واحده مكنش ليها أكيروسي كويس او كانت بعيده شويه .. انت دلوقت معاك ال setاللي هترجعلك الأقرب للصح فعلياً ..

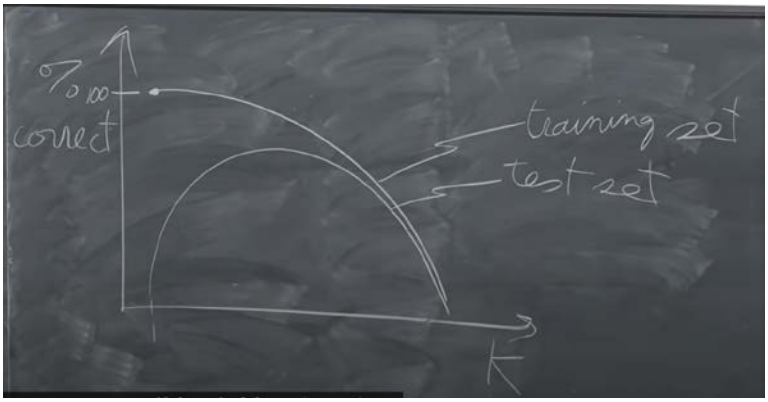
# K-Nearest Neighbour

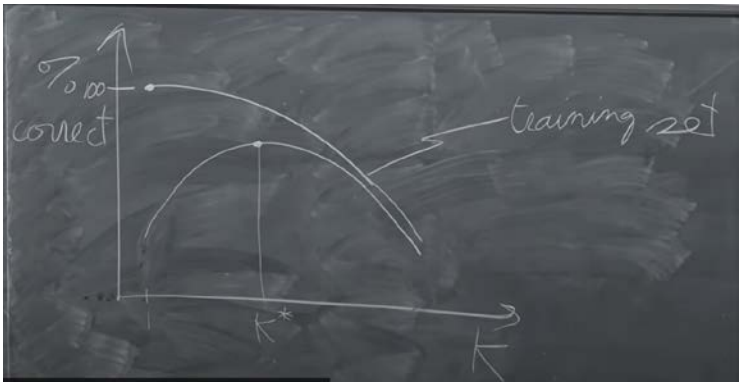
- Nearest neighbour often instable (noise)
- Idea: assign most frequent label among k-nearest neighbours
  - Let $knn(x)$ be the $k$-nearest neighbours of $x$ according to distance $d$
  - Label: $y_x \leftarrow mode(\{y_{x'} | x' \in knn(x)\})$



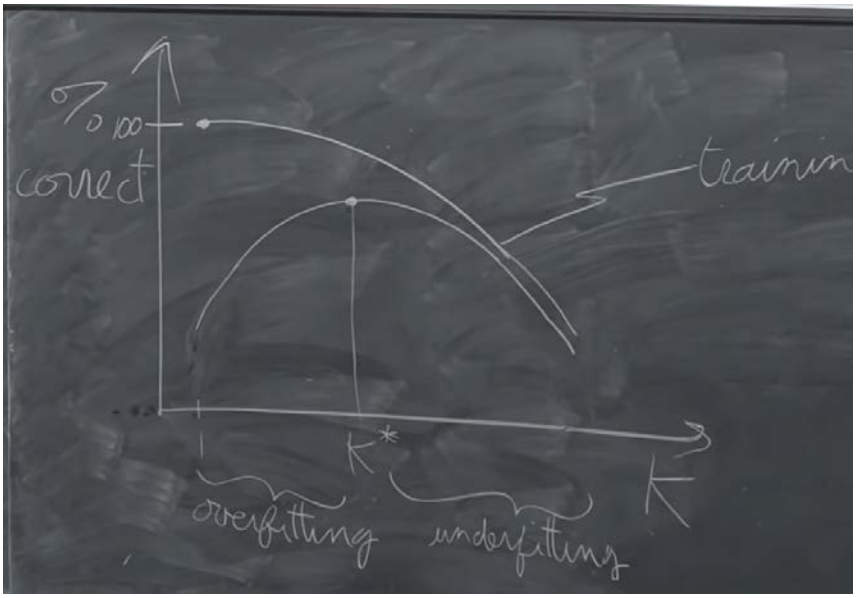
دلوقت لو انت هت consider K Nearest neighbor .. ايه هو الرقم الصحيح لل neighbors اللي عندك؟ .. ده مش حاجه سهله انك تقولها يعني .... بس جرب هي بتيجي بالتجربه ... في tradeoff... لو انت كترت ال neighoburs او انت قللتهم أوي .. هتلبس . اختار في النص .. تعال نبص علي الكيرف اللي الدكتور هيرسمه .. لو انت قلت ان عندك K = 1 ... هتلاقي عندك 100% كلهم صح .. ليه؟ .. عشان كل query هيبقا نقطه لما تلاقي ال nearest neighbor هتلاقي النقطه ديه تاني .. وتدور علي ال label هتلاقيه correct label .. وترجّعه فدايماً هترجّع ال correct label ..

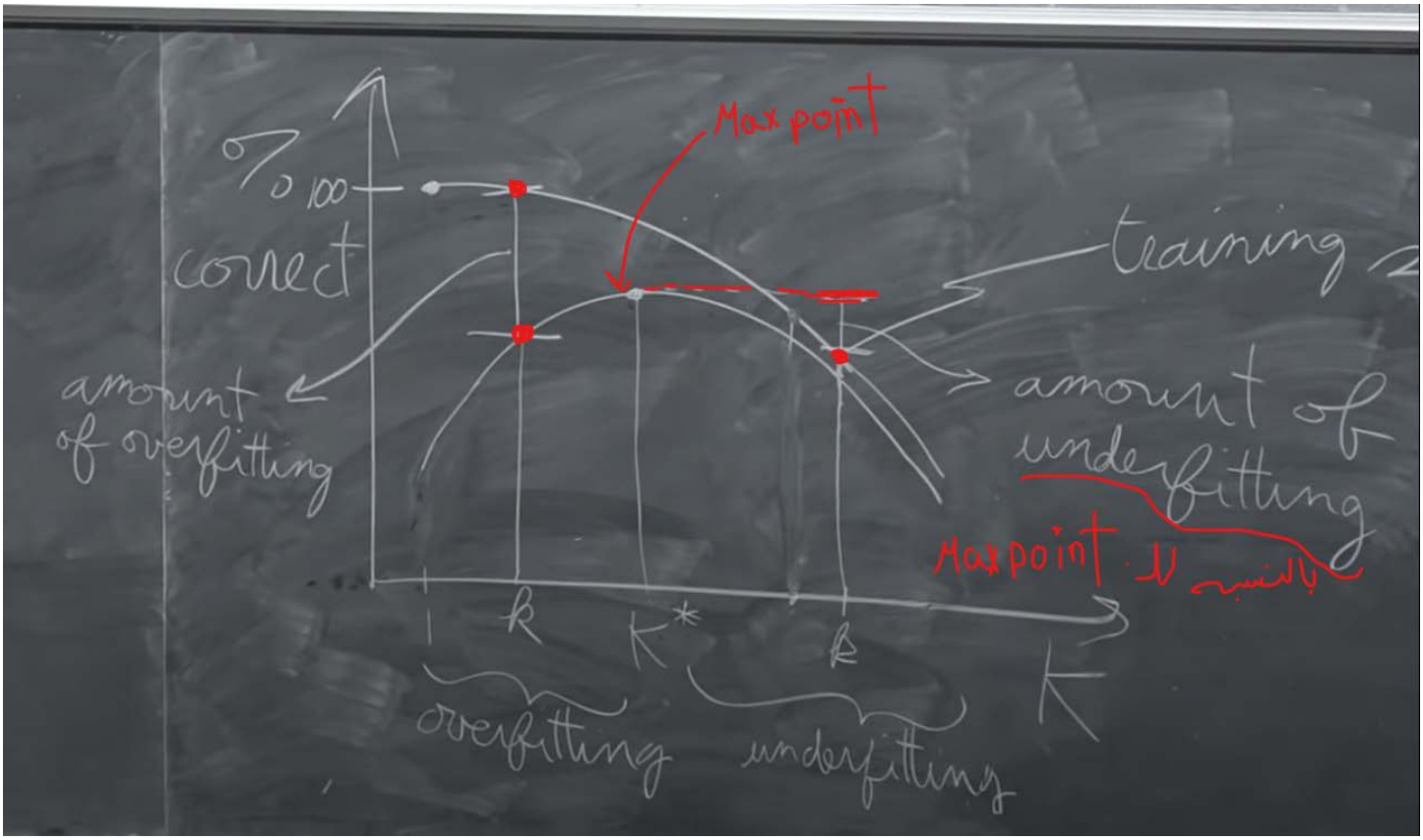
لو انت قلت عند K = 1 .. وانت بت evaluate with respect to the training set .. انت عندك ترينج سيت .. وبعد ما عملت ترينج علي الترينج سيت .. جيت تبص علي ال accuracy بتاعت ال training هتلاقيها 100% .. ليه .. عشان كل query عباره عن نقطه في الترينج سيت .. اللي هو جزء من الداتا بيز بتاعتنا .. فلما هتيجي تلاقي ال nearest neighbor هتلاقي النقطه ديه تاني .. فهترجّع ال label بتاعها فهتلاقيك جبت ال 100% لكل الداتا ... only 1 neighbor .... الفكره في كيرف كدا لل K .. كل ما ال K زادت .. كل ما الترينج سيت accuracy قلت ... و العكس بالنسبه لل test ...set




الدكتور اتكلم بشكل بسيط علي ال underfitting .. اللي هو لو في hypothesis طلعتلك accuracy سيئه جداً .. فأكيد في hypothesis تانيه بتطلع accuracy كويسه .. ال overfitting .. انت بت fit الداتا جامد جداً .. اللي هو كمان انك بت fit النويز اللي في دالداتا جامد جداً برضو .. فانت بتضرب ال test accuracy .. في السلايد هنا بيقولك ال futureAccuracy هنا هي الداتا الي ممكن تجيلك قدام اللي هي ال test accuracy .. الدكتور رجع للكيرف اللي فوق ده تاني عشان يوضح عليه ال overfitting و ال underfitting ...






# Choosing K

- How should we choose K?
  - Ideally: select K with highest future accuracy
  - Alternative: select K with highest test accuracy
- **Problem:** since we are choosing K based on the test set, the test set effectively becomes part of the training set when optimizing K. Hence, we cannot trust anymore the test set accuracy to be representative of future accuracy.
- Solution: split data into **training, validation and test sets**
  - **Training set:** compute nearest neighbour
  - **Validation set:** optimize hyperparameters such as K
  - **Test set:** measure performance



سؤال جامد تاني .. ازاي نختار K ... لو انت اخترت ال k بناءً علي الtest accuracy .. انت كدا بتعمل ترينج علي التست سيت مش علي الترينج سيت .. لإنك كدا كإنك effectively بتعمل training on test set .. فال K هنا هو بارمتر إنتا بتعملو optimization .. وبكدا انت بتعملو optimization باستخدام ال test set .. وبكدا انت كإنك واخد في الاعتبار ال test set كإنها في الترينج داتا .. اومال نعمل ايه .. split into 3 sets .. ترين و فاليديت و تست .. ال validation هتساعدك انك ت optimize the hyper parameter .. وهو ده الهدف من ال validation set ..تعال نبص علي الصودو كود مش الصودو كريم بتاع ال انك تختار ال K

# Choosing K based on Validation Set

Let $k$ be the number of neighbours

For $k = 1$ to max # of neighbours

$h_k \leftarrow train(k, trainingData)$

$accuracy_k \leftarrow test(h_k, validationData)$

$k^* \leftarrow argmax_k\ accuracy_k$

$h \leftarrow train(k^*, trainingData \cup validationData)$

$accuracy \leftarrow test(h, testData)$

Return $k^*, h, accuracy$



لو انا اخترت ال K بناءً علي how well our hypothesis performs with respect to to our validation set .. لو هي اصلا صغيره و فيها نويز .. ممكن هوب دابل كيك حظ كدا .. تلاقي واحده هي احسن قيمه لل K .. بس هو حظ عشان كان في شوية نويز علي الداتا بتاعت ال validation set .... ازاي نخلي الاختيار more robust؟ .. انت تقدر تزود ال size of validation set .. ده هيحل المشكله بس هيظهرلك مشكله تانيه .. انك هتقلل الداتا بتاعت التريننج .. فاتضح إن في طريقه تقدر تستخدم ال training and validation set الاتنين للاتنين ال training and validation و ده اللي هو ال cross validation .. ده بيحصل عن طريق انك تاخد الداتا . وتقسمها نصين .. وتفضل تقسم تقسم تقسم تقسم ... وبعدين تاخد ال average accuracy across all the experiments with respect all those splits .. فانت كدا هتعمل كدا K times ... اللي هي K-fold cross validation

## Robust validation

- How can we ensure that validation accuracy is representative of future accuracy?
  - Validation accuracy becomes more reliable as we increase the size of the validation set
  - However, this reduces the amount of data left for training
- Popular solution: **cross-validation**

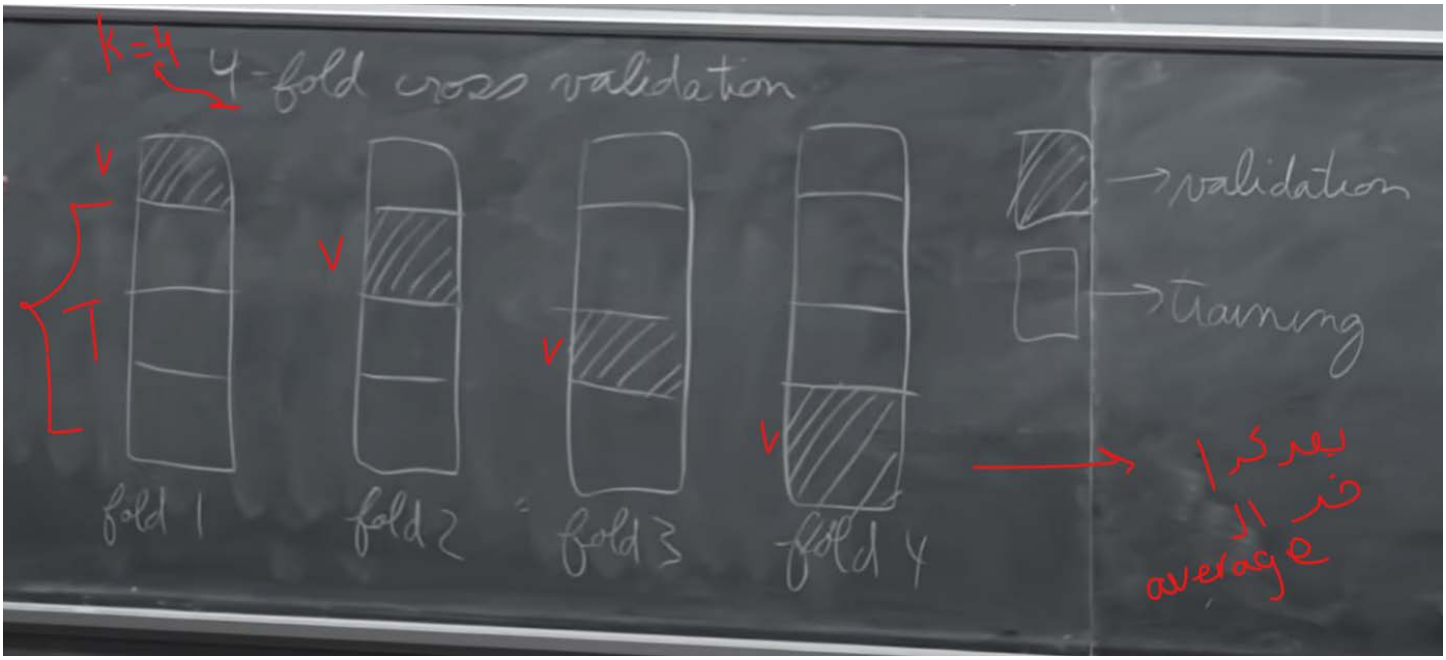


## Cross-Validation

- Repeatedly split training data in two parts, one for training and one for validation. Report the average validation accuracy.
- ***k*-fold cross validation**: split training data in $k$ equal size subsets. Run $k$ experiments, each time validating on one subset and training on the remaining subsets. Compute the average validation accuracy of the $k$ experiments.
- Picture:



تعال نرسم صوره تكون أوضح شويه للمعني ...


k=4
4-fold cross validation
V
T
V
V
V
fold 1
fold 2
fold 3
fold 4
validation
training
يعد كل
حد ال
average