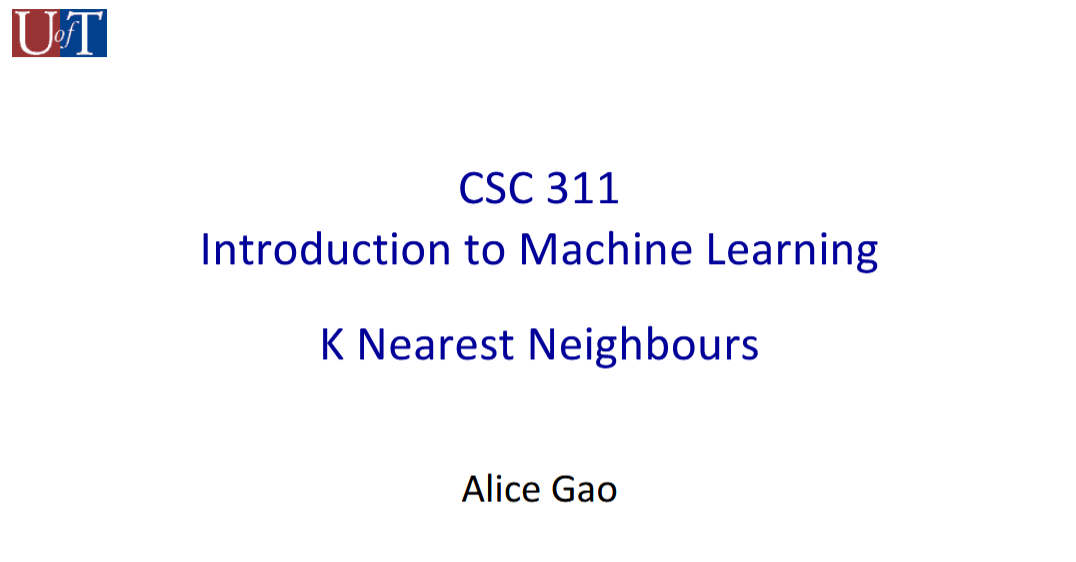
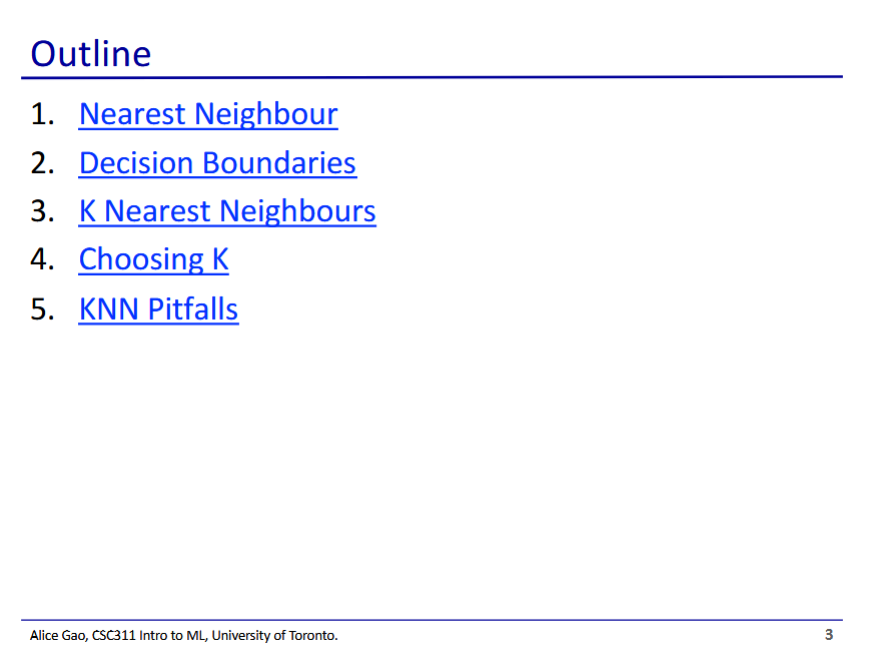
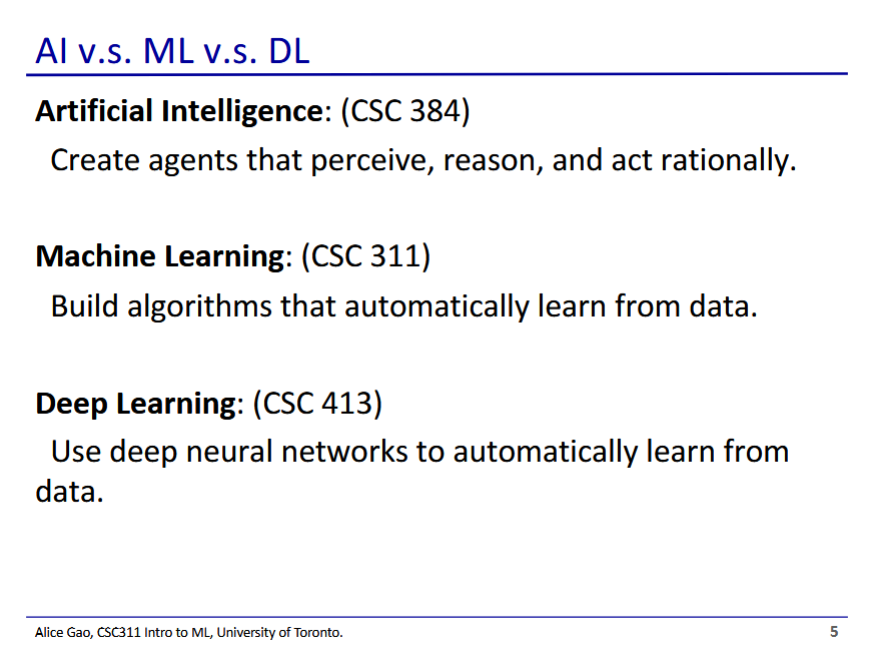
| **What is ML?**   * Field of artificial intelligence involving creating algorithms that learn to solve a problem by training on data * Types of machine learning problems   + **Supervised learning**     - Training data contains input values paired with correct output values     - Algorithm predicts labels given inputs     - Most of our course will handle supervised learning   + **Unsupervised learning**     - Training data contains unlabelled data     - Algorithm looks for patterns in the data   + **Reinforcement learning**     - Somewhere in-between supervised and unsupervised learning     - Periodically gives a reward signal to the algorithm, algorithm tries to maximise reward   **Supervised learning**   * Process by which the algorithm learns a function that maps an input to an output * Data:   + Input data:     - Each input data is represented by a D-dimensional vector ()     - Each component of the vector (= ith component) represents some characteristic of the input       * Also called a **feature**       * In an image input, the intensity value of each pixel would each be a feature   + Output data:     - Output data is a scalar value (t)     - In a regression problem, the output is continuous     - In a classification problem, the output is discrete * **Training data vs test data**   + We split up our labelled data into 2 parts: training data and test data   + **Training data** is used to train the model to learn the function     - the i'th input value of training data       * the j’th feature of the i’th input value of training data     - the i'th output value of training data   + **Test data** is used to evaluate the model     - Algorithm predicts labels given the input data     - We don’t want to evaluate the model using the training data since the model is already fitted to that data       * We want to see how our model works with new unseen data   **Evaluating classifier models**   * We can evaluate our model by running it on the test set (data not seen by model before) * **The test set cannot be used to tune hyperparameters (such as k in kNN)**   **Nearest neighbour algorithm**   * Supervised ML algorithm to classify objects * **Process:**   + Compute the distance between new data and all training data and find the training data point closest to it   + The predicted label of the new data is the label of the closest training data point () * Distance can be measured in many different ways   + Euclidean distance (default, measures distance intuitively)   + Cosine similarity (measures angles between vectors)   + Etc. * **Decision boundaries**    + Regions (lines) where the output of the classifier is ambiguous     - Output changes as you cross a decision boundary   + Can be visualised using a **Voronoi diagram** * **Problems:**   + Nearest neighbour is sensitive to noise/mislabelled data     - Each isolated data point in the training data creates a pocket around it (see Voronoi diagram)   + Thus, it may overfit to the training data   **k nearest neighbours algorithm**   * Like nearest neighbour, but instead uses the **k nearest points** to determine label   + A higher k makes smoother decision boundaries with fewer isolated regions     - Too high k and the model will underfit   + A low k results in the model focusing more on detail in the data     - Too low k and the model will overfit   + Setting the value of k involves making a tradeoff between the two * **Choosing k**   + k is a **hyperparameter**- a setting we set outside of the learning algorithm   + Optimal choice of k depends on the data and its size     - Typically we chose   + Alternatively k can be chosen using a **validation set**     - We set aside data to use to test out which k is best to use       * The k value with the best performance on the validation set is used     - **Validation set must be separate from the test data**, the test data can only be used once at the end to evaluate the model |
| --- |

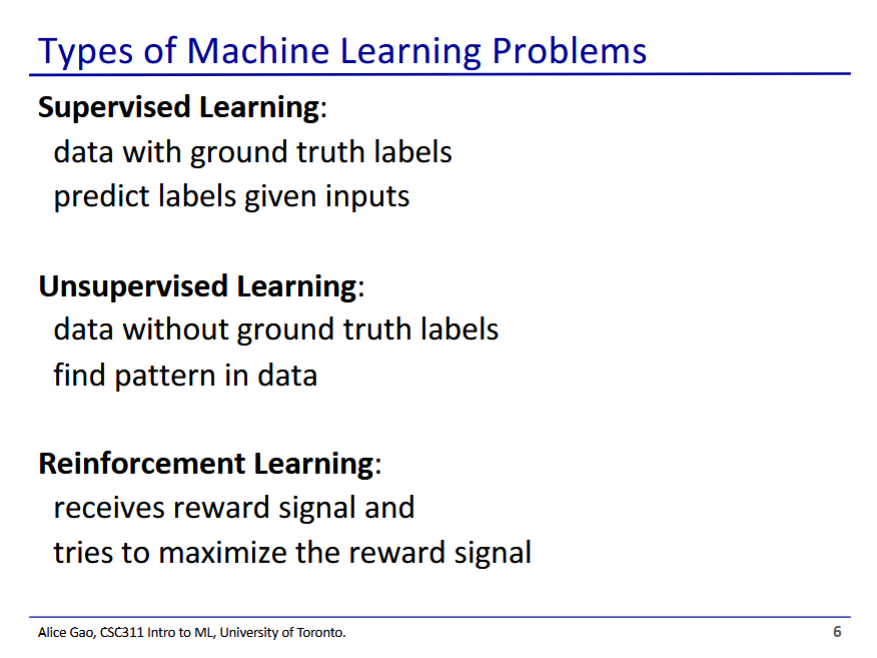




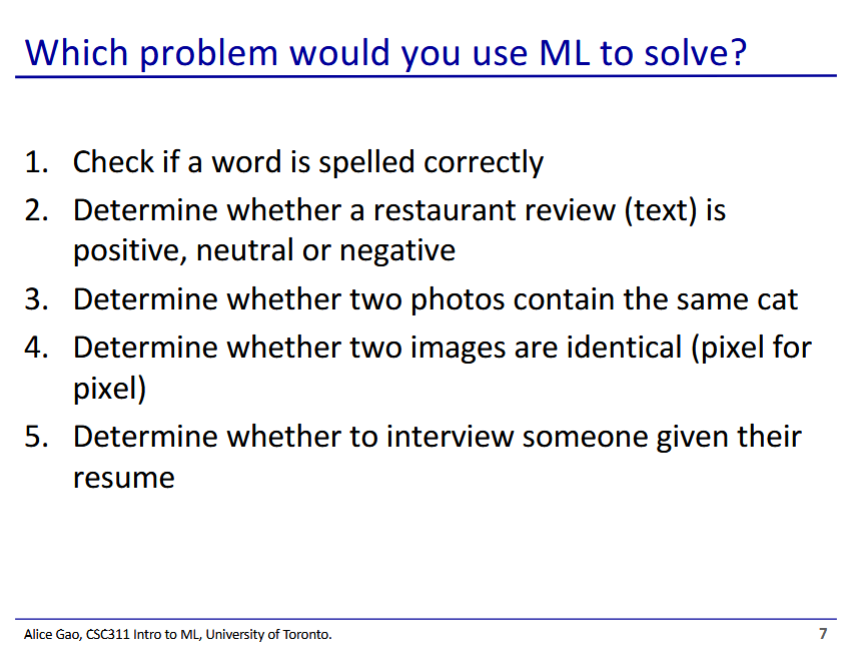




* Machine learning
  + Is only one field in AI
  + Involves using data to train an algorithm to solve a problem
* Deep learning
  + Neural network model with many layers of nodes



* Supervised learning
  + Pairs correct inputs with correct outputs in the training data
  + Most of our course will handle supervised learning
* Unsupervised learning
  + Training data does not contain the “correct answer”
  + Algorithm looks at data and tries to find patterns by itself
* Reinforcement learning
  + Somewhere in-between supervised and unsupervised learning
  + Periodically gives a reward signal to the algorithm, algorithm tries to maximise reward



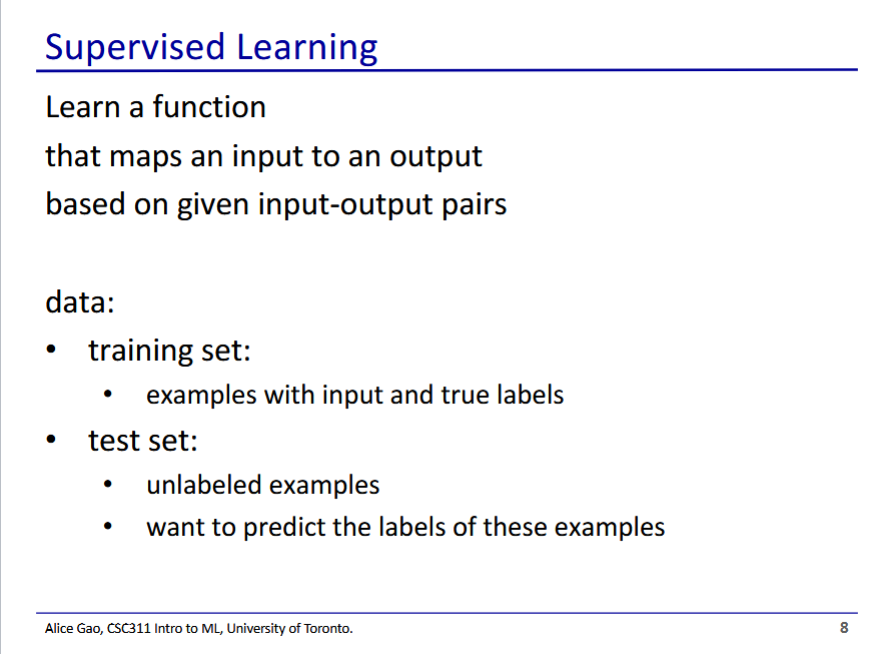
1. No

* Can compare words with a set of correctly spelled words

1. Yes
2. Yes
3. No

* Comparing images pixel by pixel is a simple problem that doesn’t need ML

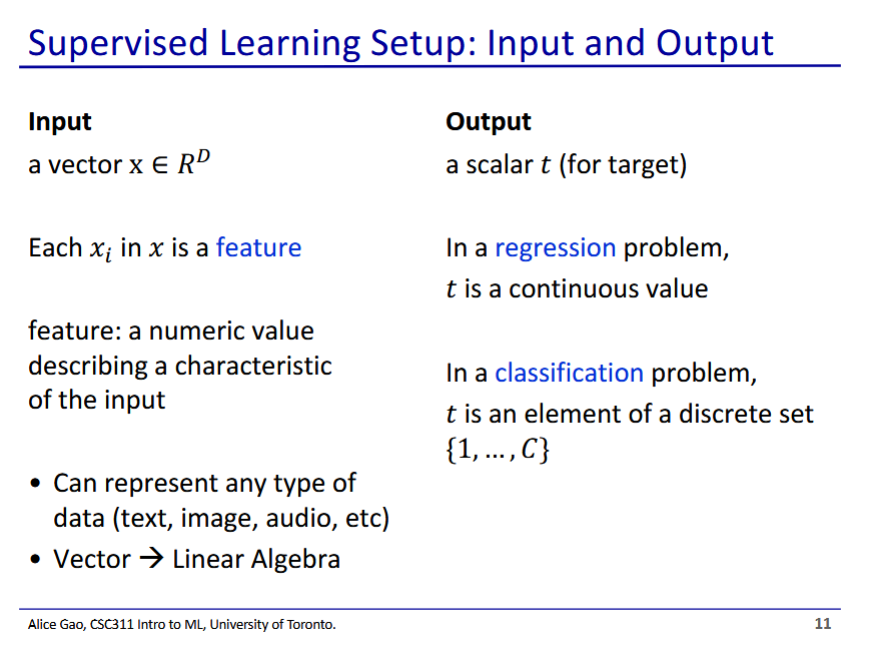
1. Yes?



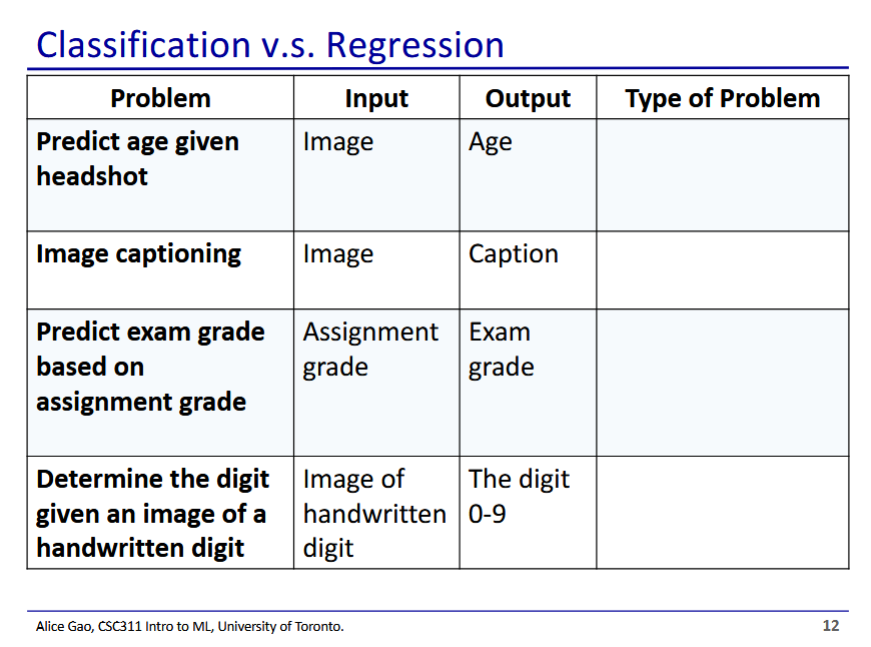
* Typically, we divide our data into 2 parts:
  + Training set - data used to train the model
  + Test set - data used to evaluate the model
  + We don’t want to evaluate the model using the training data since the model is already fitted to that data
    - We want to see how our model works with new unseen data



* Predict age given headshot
  + Input: headshots of people
  + Output: age of person in image
* Image captioning
  + Input: image
  + Output: caption

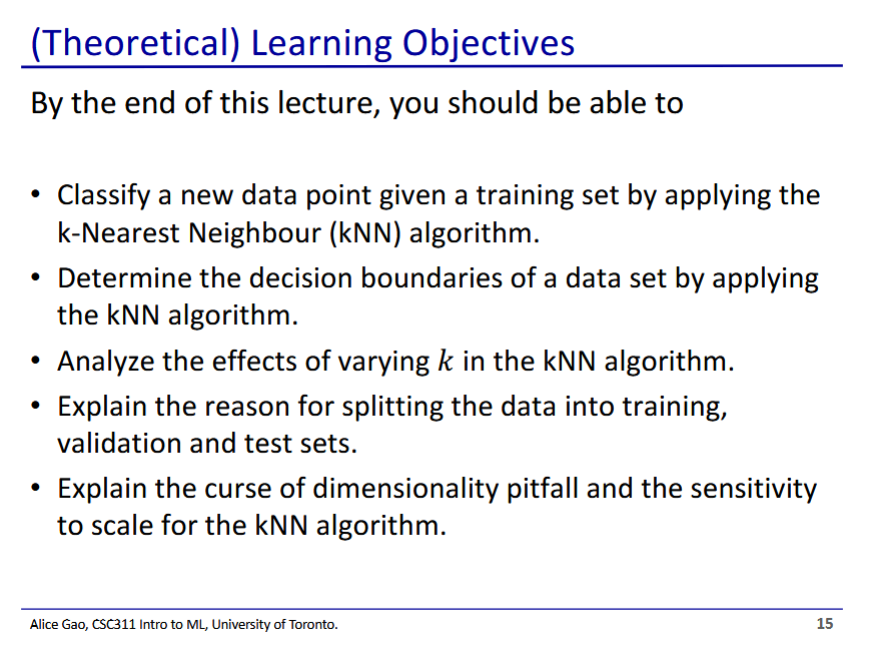


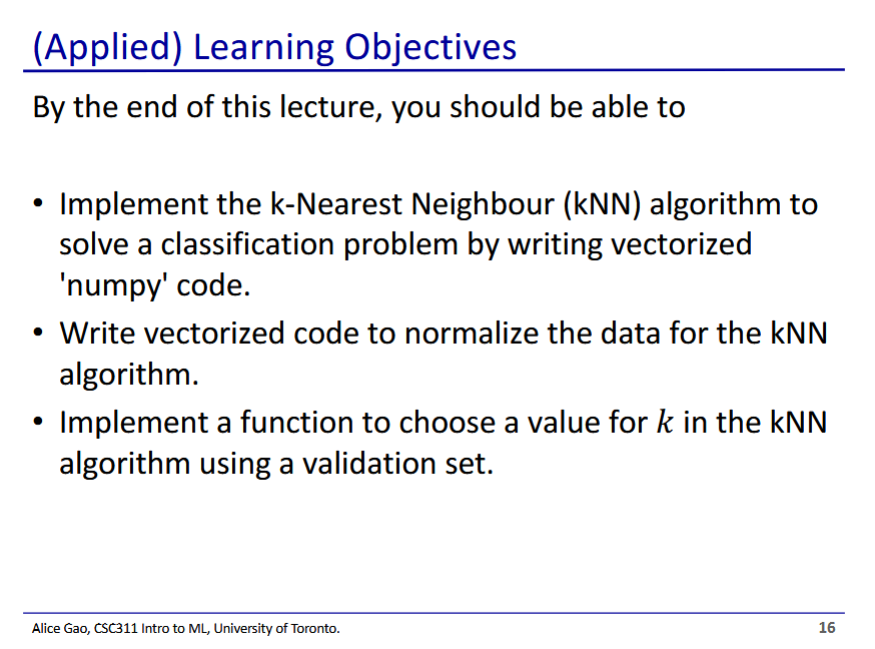
* Input is given as a D-dimensional vector (x)
  + Has D components (xi) of the input vector which are features
  + Now that input is a vector, we can do linear algebra on it
* Regression problems have continuous outputs
* Classification problems have discrete outputs

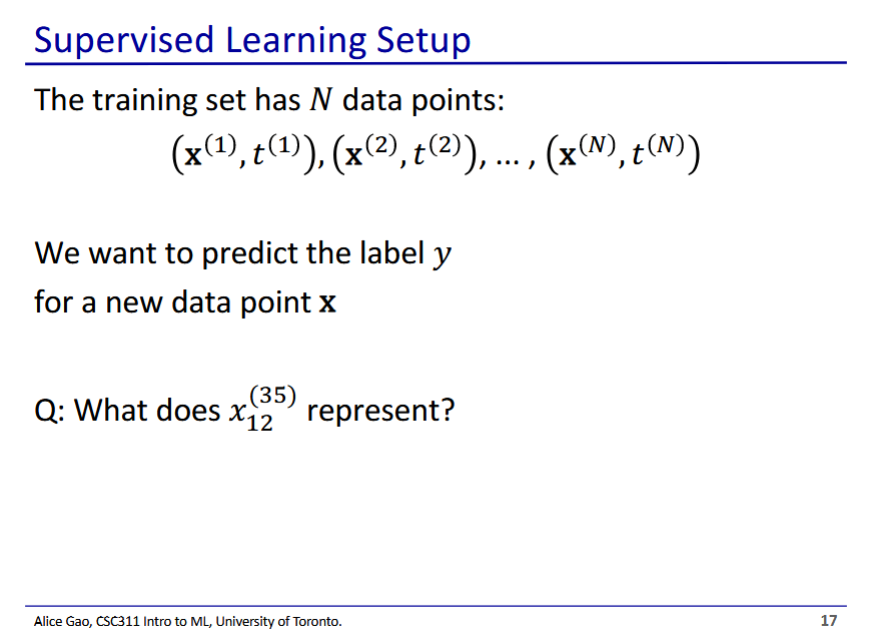


* Both could work
* Regression
* Regression
* Classification

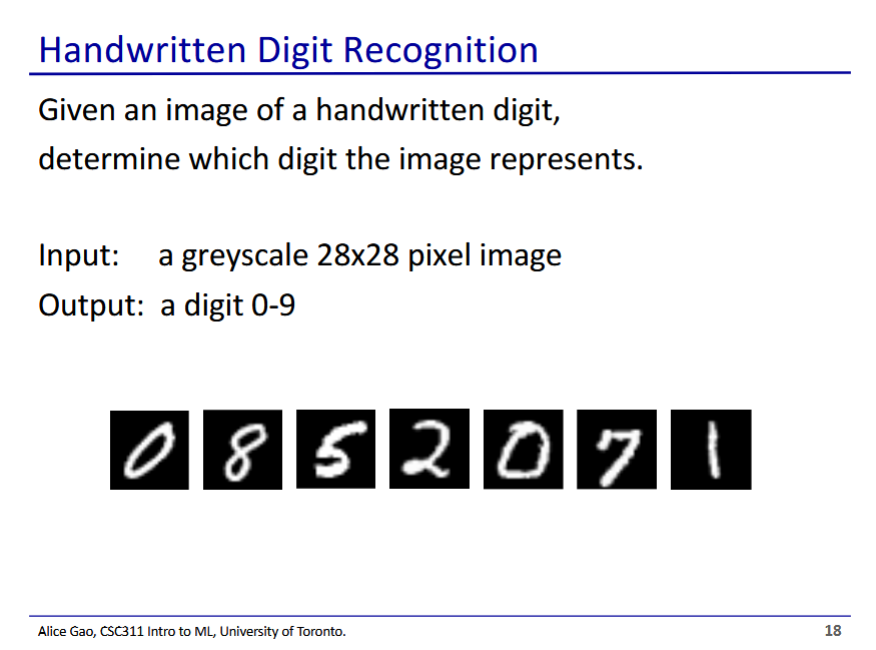


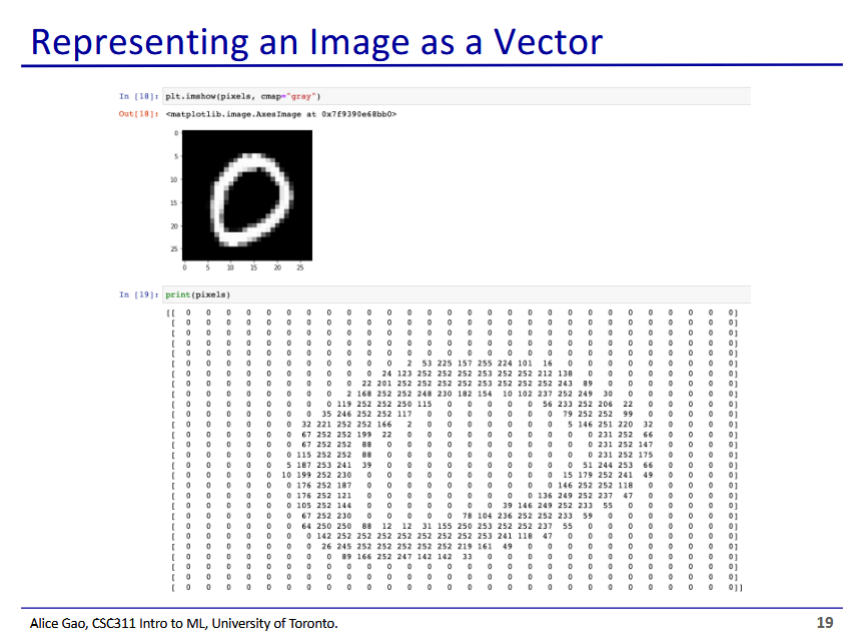




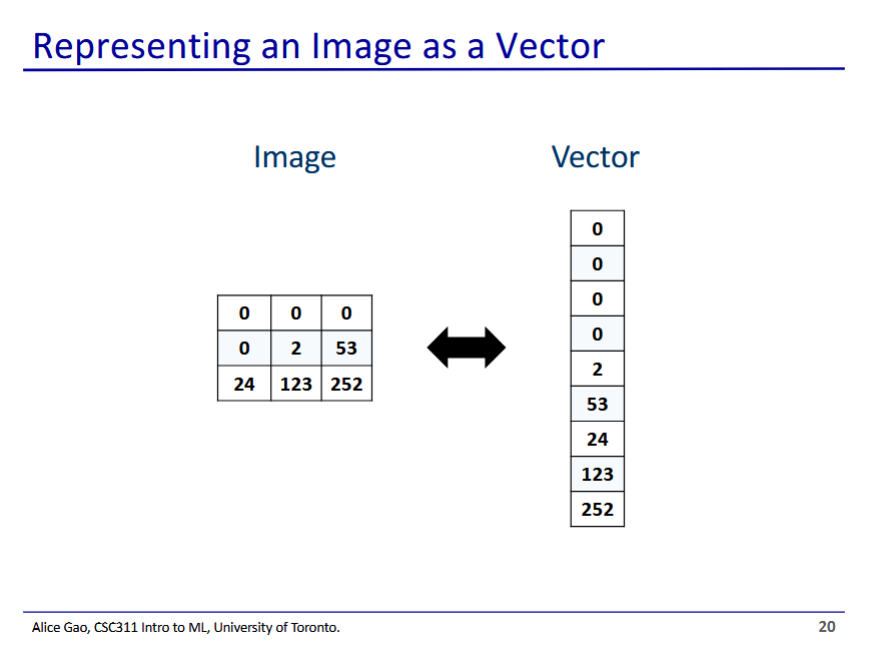


* Training set
  + Superscript indicates the index of the training data entry
* What does represent?
  + The 12th feature of the 35th data point’s input

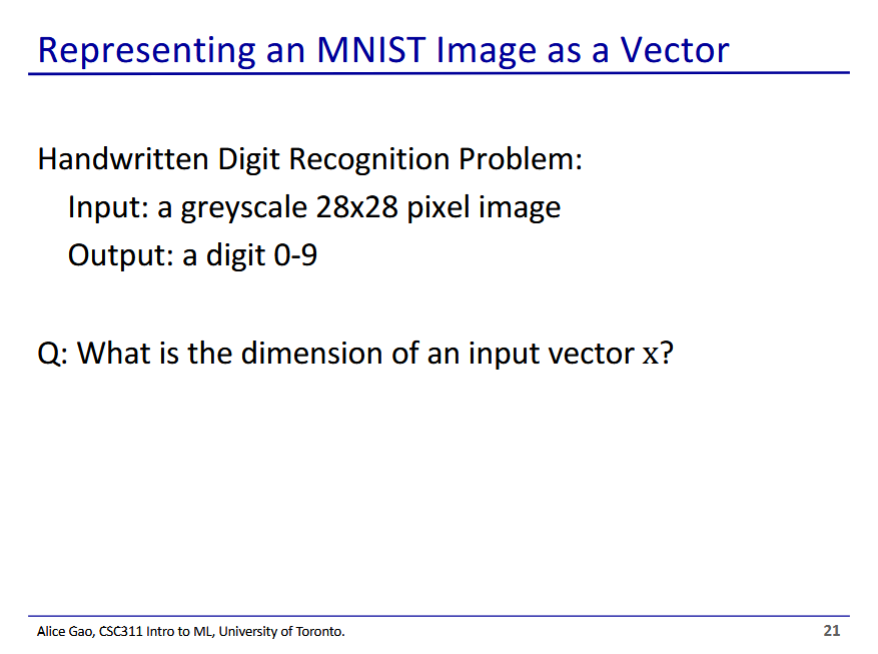




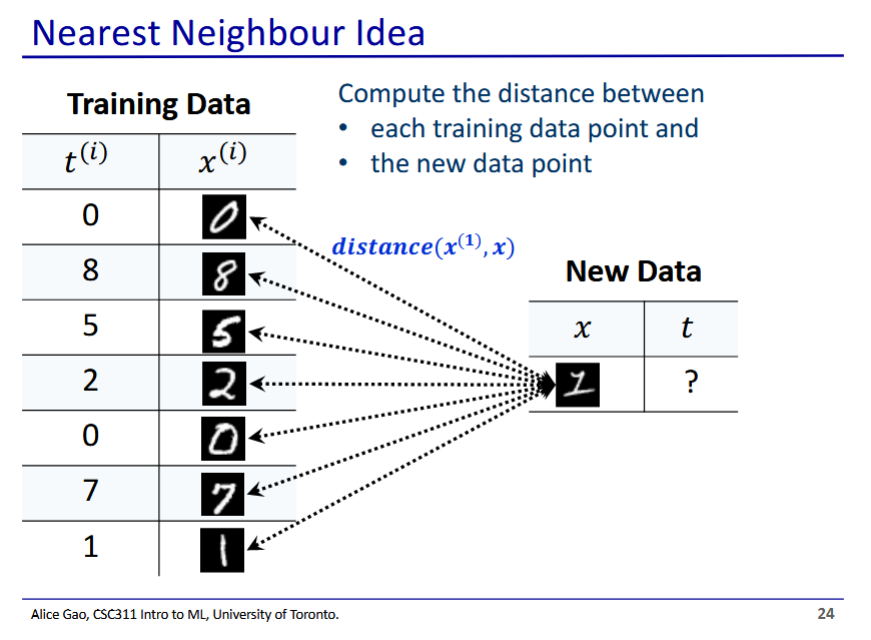
* An image is a matrix of pixels each with an intensity value



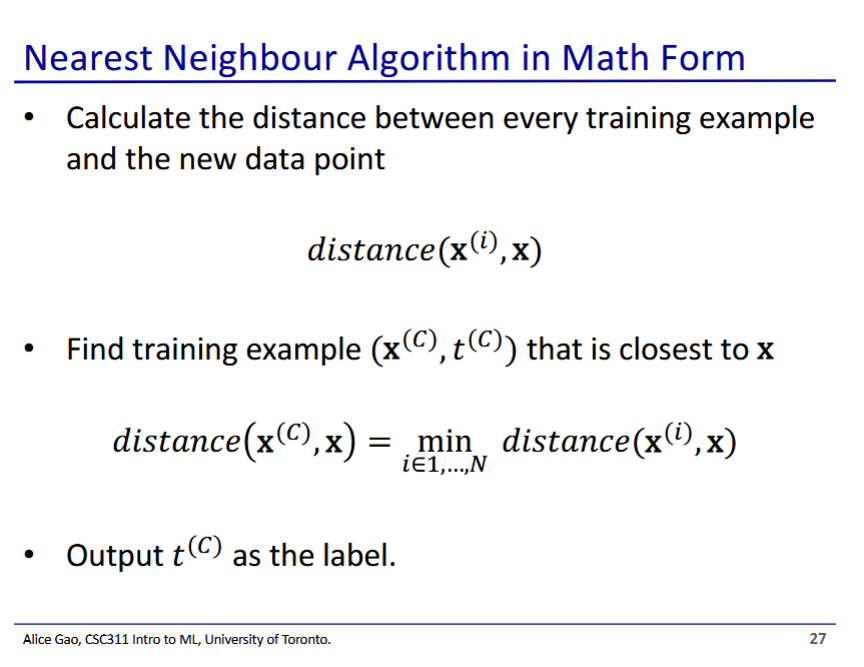
* We can flatten it out to make a vector
  + Each feature is the intensity of 1 pixel in the image

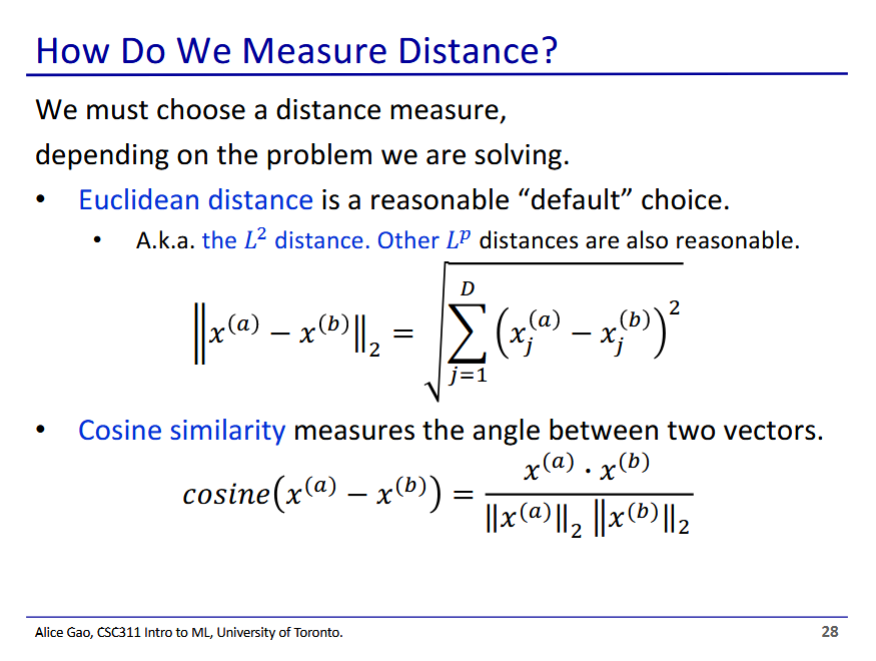


* The dimension of input vector x is

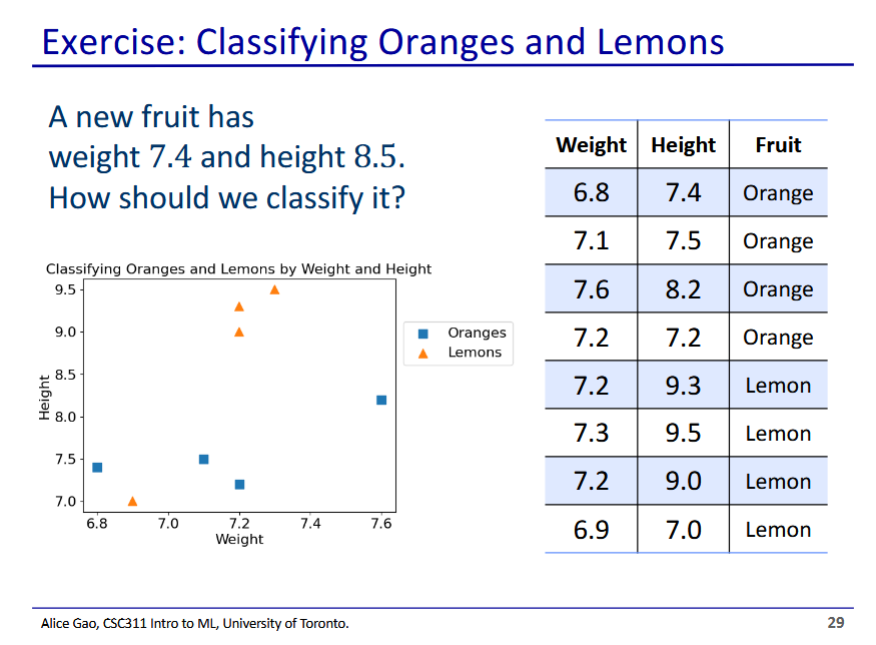


* Nearest neighbour (not k-nearest neighbour yet)
  + We calculate the distance between our new data, and every point in our training data
    - We try to find the closest training data point to our new data
  + The prediction is the identity of the closest training data point

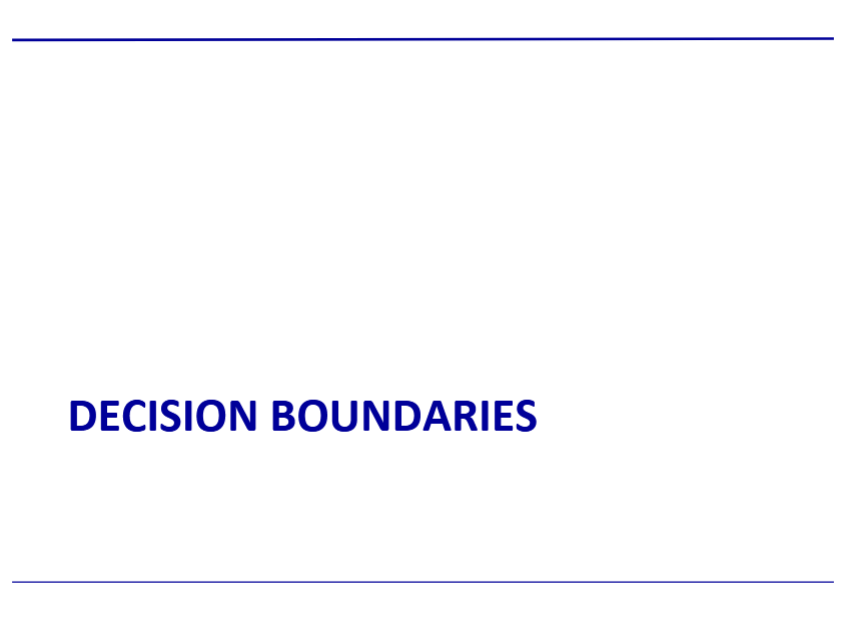


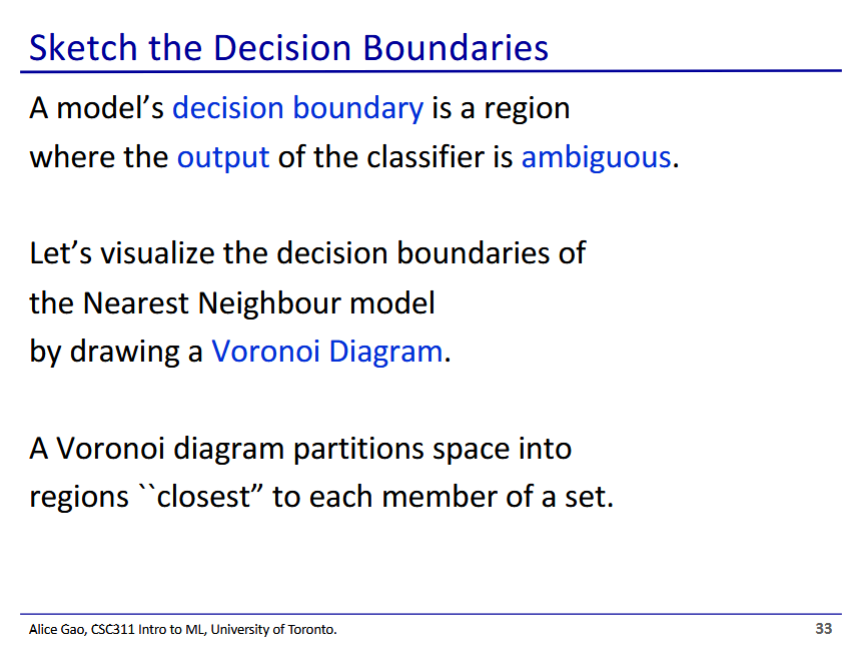


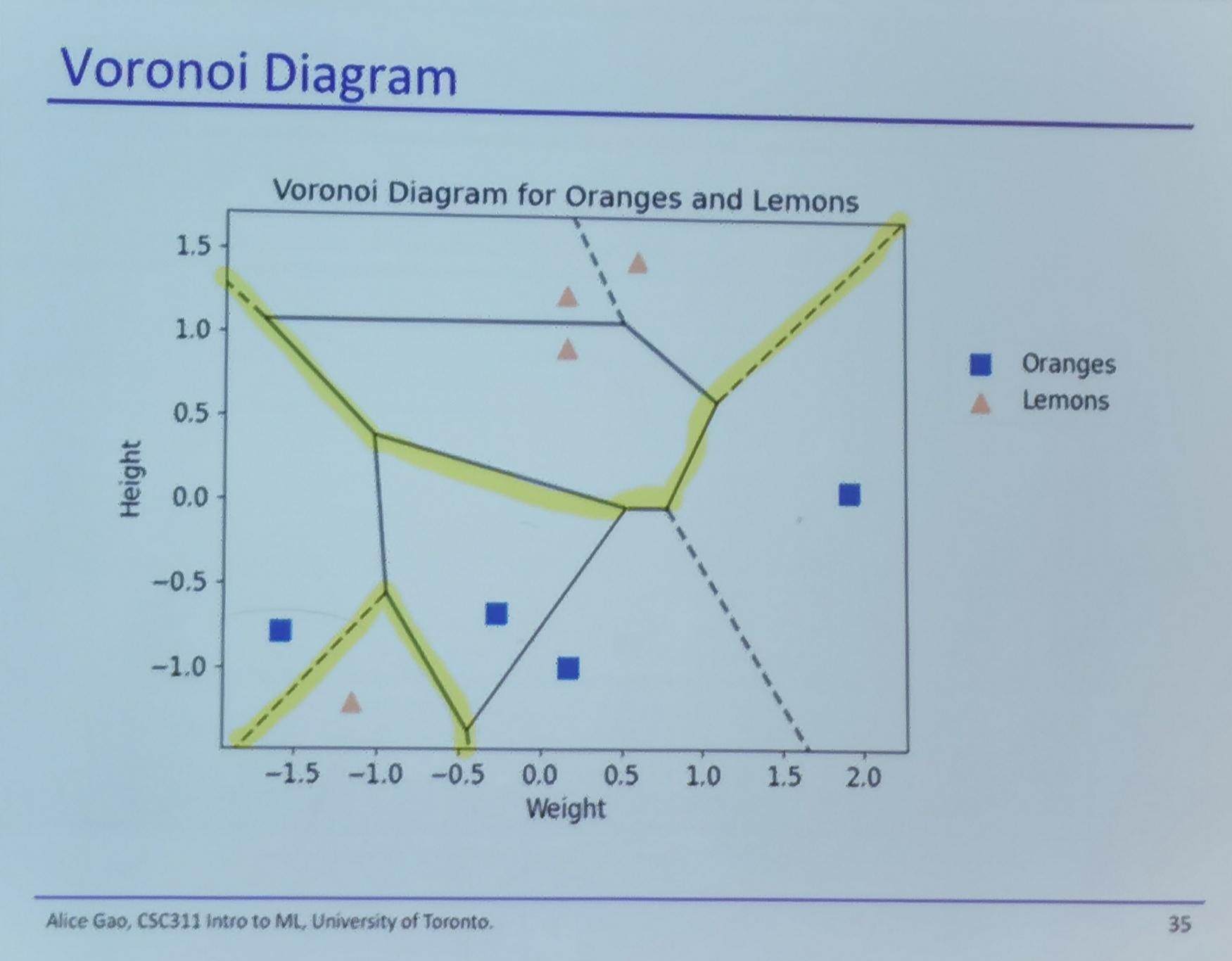
* We have many possible choices for how to measure distance between data points

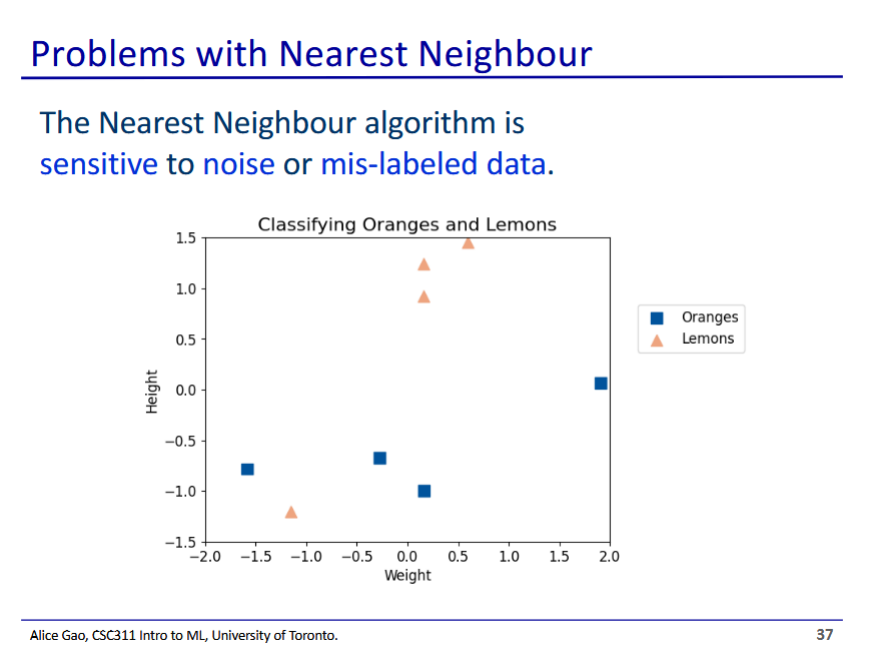


* We can plot the new fruit on the graph, and find the closest fruit to it (orange)
  + If we were to be fully faithful to the algorithm, we would need to look at the distance to every training data point from our new data point

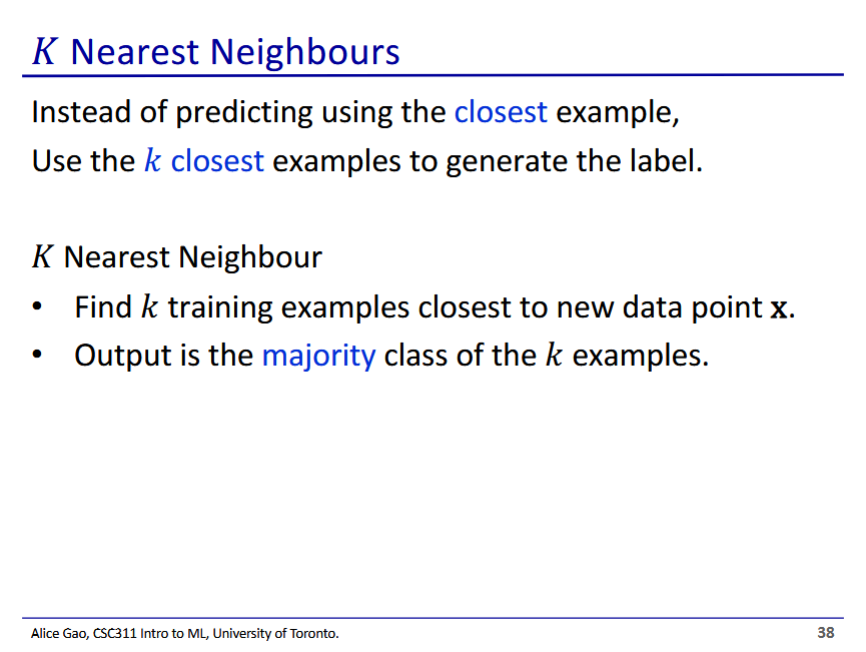




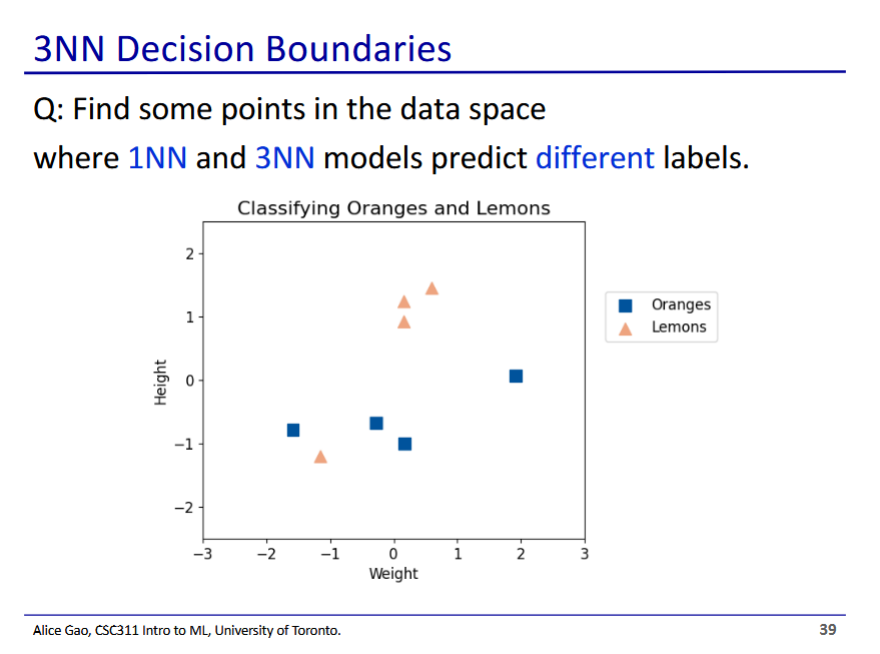




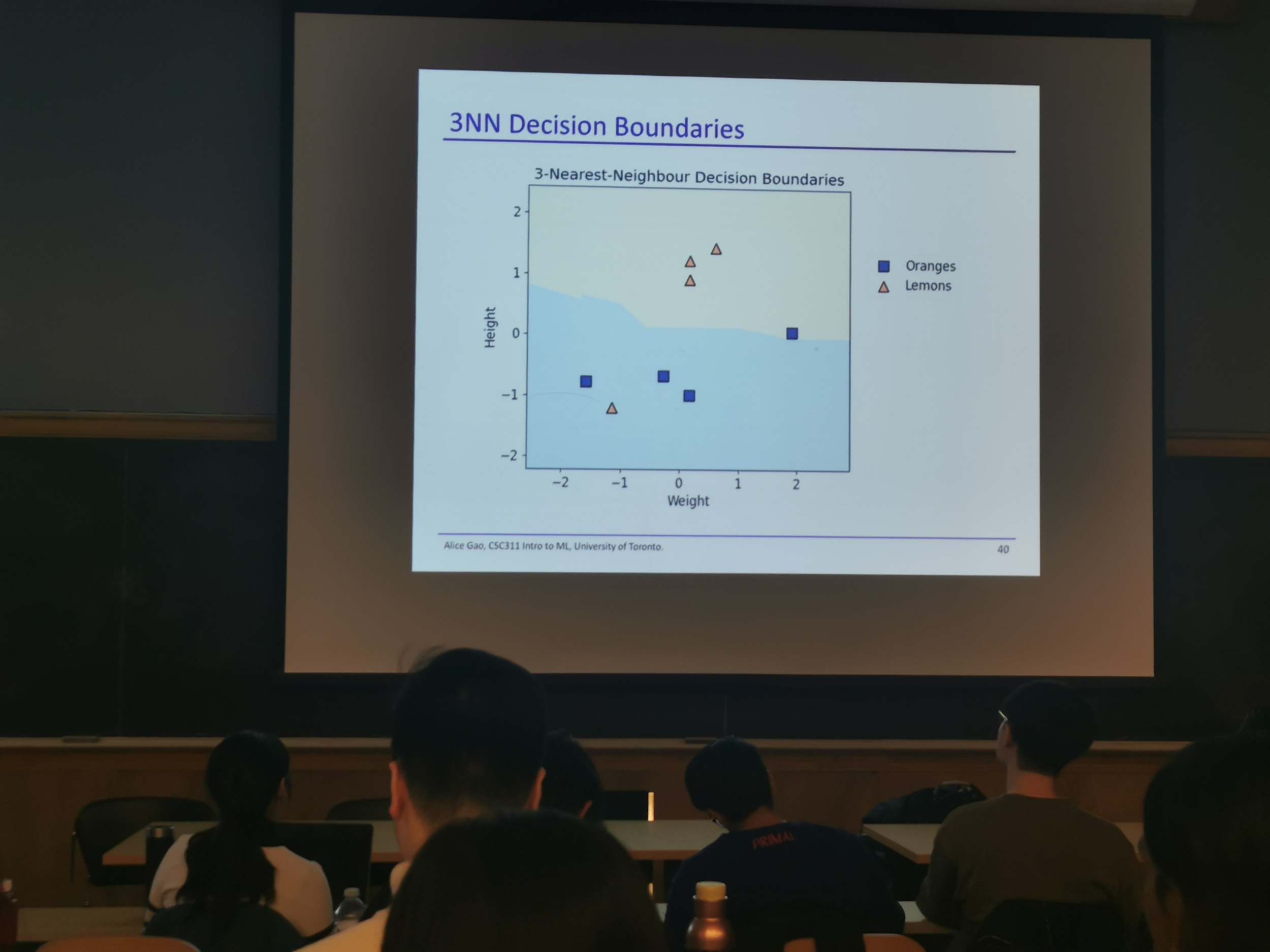
* Say if the bottom lemon was mislabelled or due to noise, the entire region around that lemon would be labelled lemons
  + This makes the model more sensitive to noise or mislabeled data

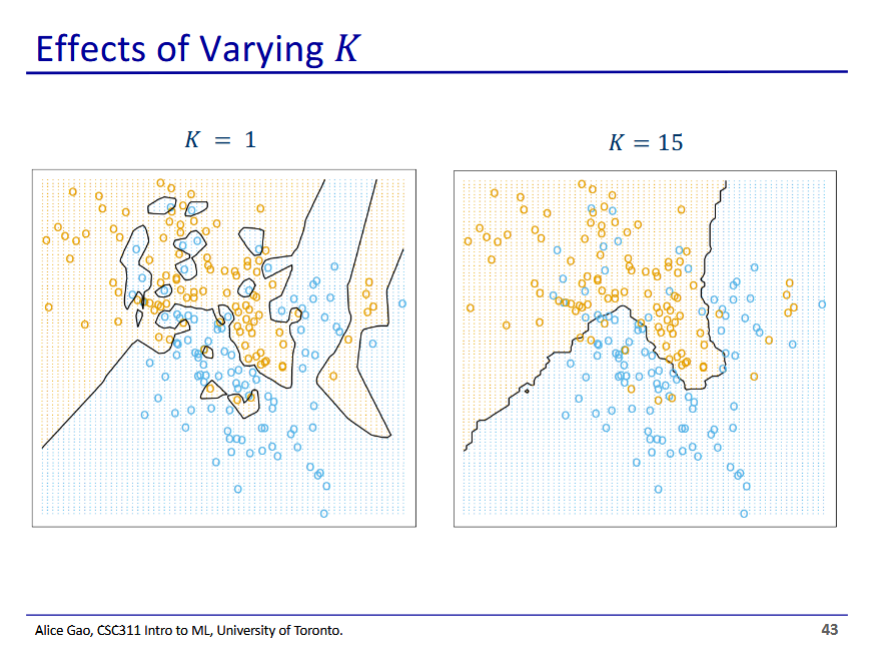


* We set k to whatever integer you want
* We use the identities of the closest k neighbours to identify our new data

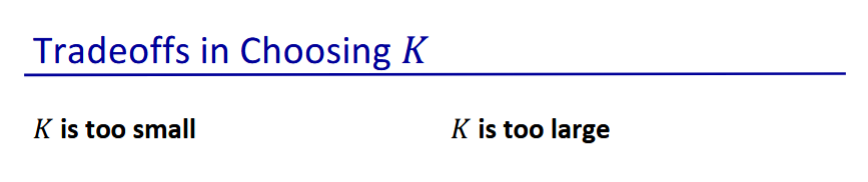


* Near the isolated orange and the isolated lemon
  + In these regions, the single closest lemon/orange would overpowered by the different fruits nearby

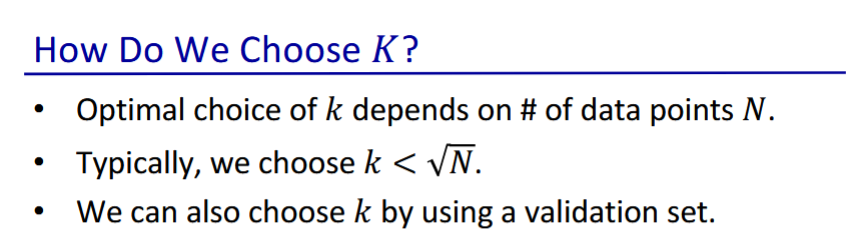




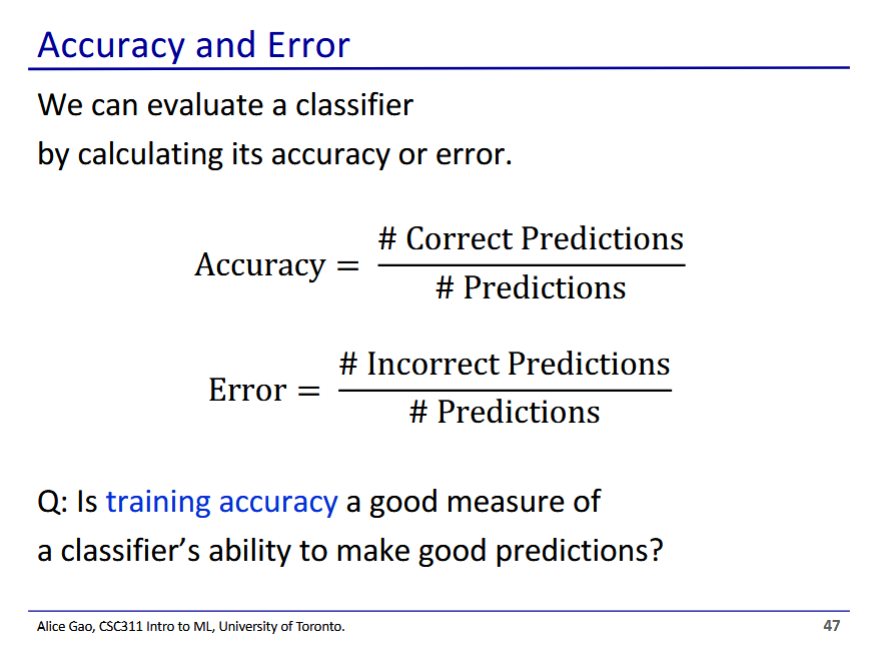
* Higher values of k make for smoother decision boundaries with fewer isolated regions
* A low k results in the model focusing more on detail in the data
* There is a tradeoff when choosing k value



* When K is small
  + Captures small details in pattern
  + But not all details are important (could be noise)
  + May overfit to the data
* When k is large
  + Makes stale predictions
  + May overlook small details in pattern
  + May underfit to data



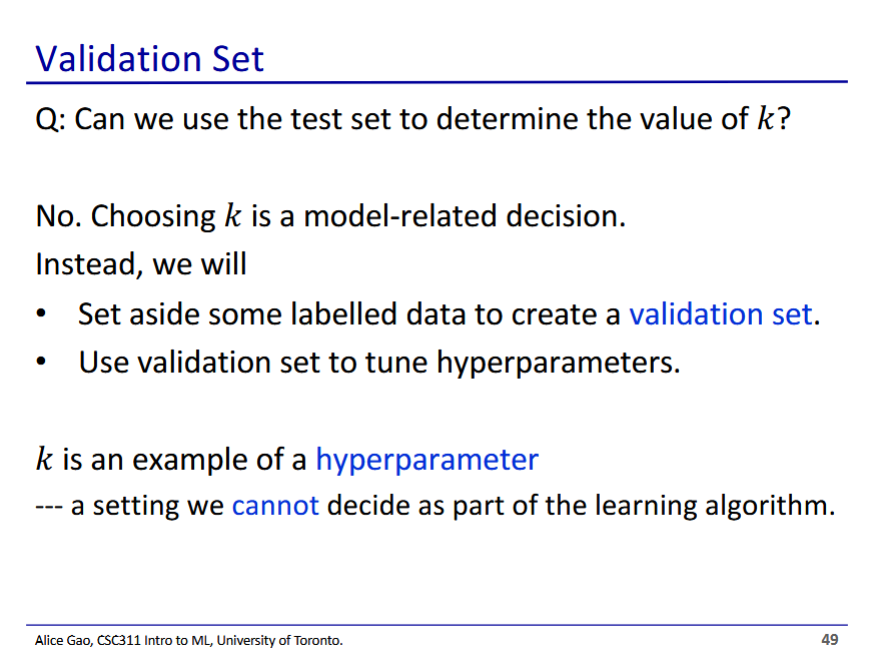
* Validation set is different than the test set

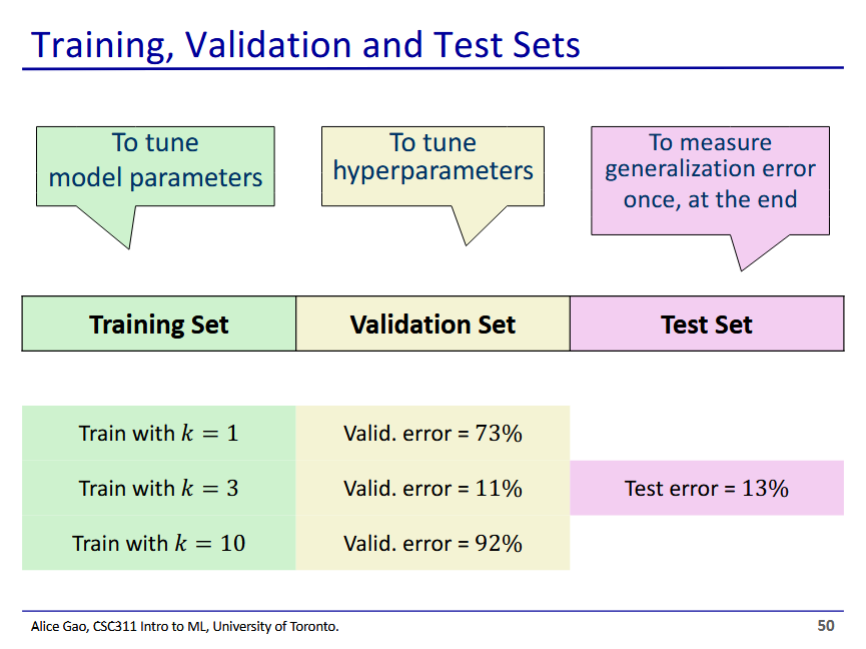


* No, we still need to evaluate the model using the test data (model could have overfit to training data, producing high accuracy on training data)



* We should not use the test set to turn hyperparameters (such as k)
* We only use the test set once at the very end to evaluate the model’s performance





* Validation set is used to test which k value is best
  + In this example, k=3 gives the smallest error
* Once we have determined our k value, we can test our final model using the test set