GENERAL ASSEMBLY

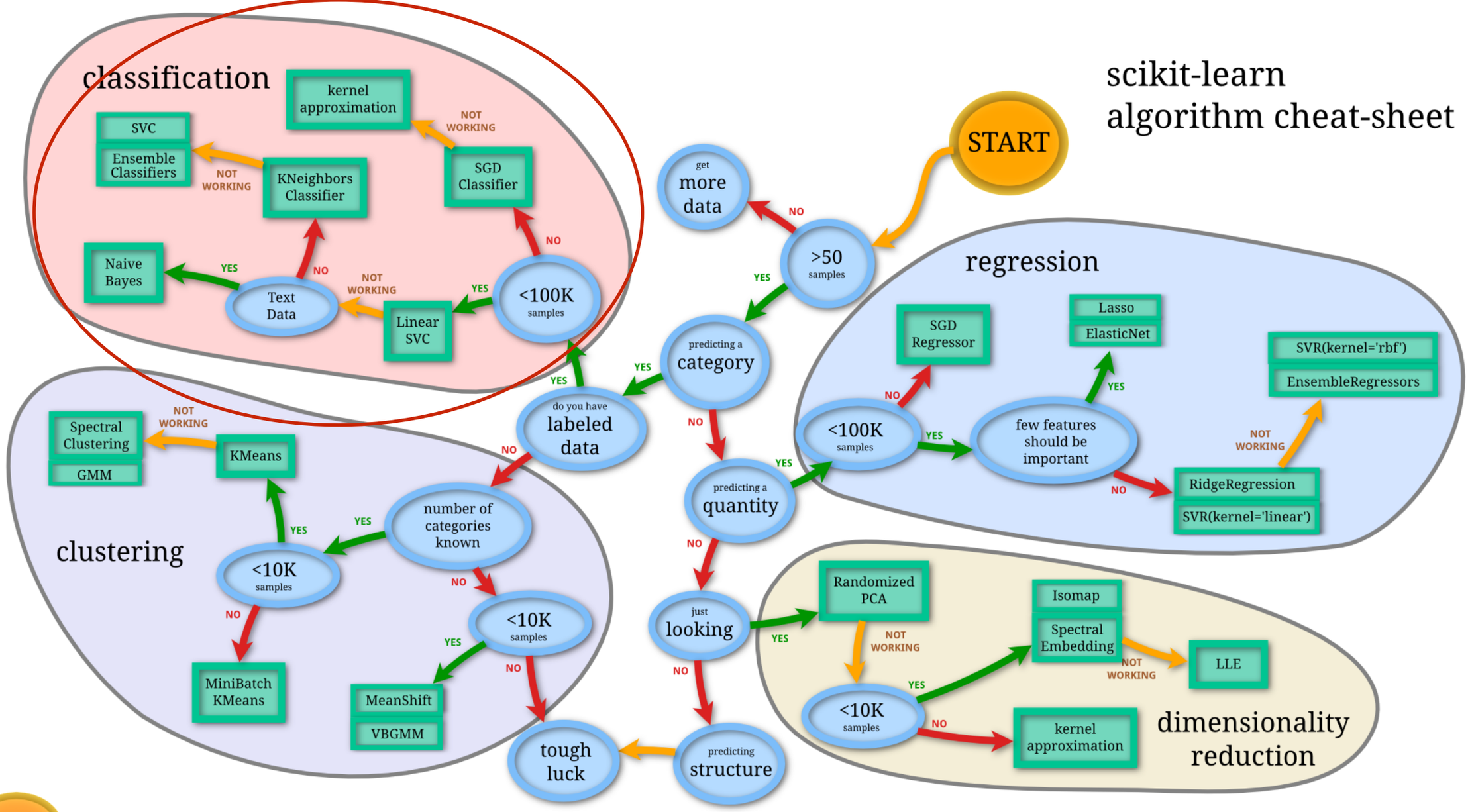
INTRO TO CLASSIFICATION

LEARNING OBJECTIVES

By the end of this lesson, we will be able to:

- Define class label and classification
- Compare and contrast regression and classification
- Code a basic classifier using the Iris dataset
- Build a K-Nearest Neighbors classifier using the sci-kit-learn library

scikit-learn algorithm cheat-sheet



Back

scikit learn

INTRO TO CLASSIFICATION

- Any class guesses?

ACTIVITY: KNOWLEDGE CHECK

EXERCISE

IN SMALL GROUPS, ANSWER THE FOLLOWING QUESTIONS

1. How have we been using linear regression?
2. How have we been evaluating our linear regression models?
3. Do you think we can use linear regression to identify gender? What about hair color? Education level?
4. Would the same methods and metrics apply to the kind of problem in (2.)?

DELIVERABLE

As a group, come to a consensus. Each group will answer one of these questions.

WHAT IS CLASSIFICATION?

WHAT IS CLASSIFICATION?

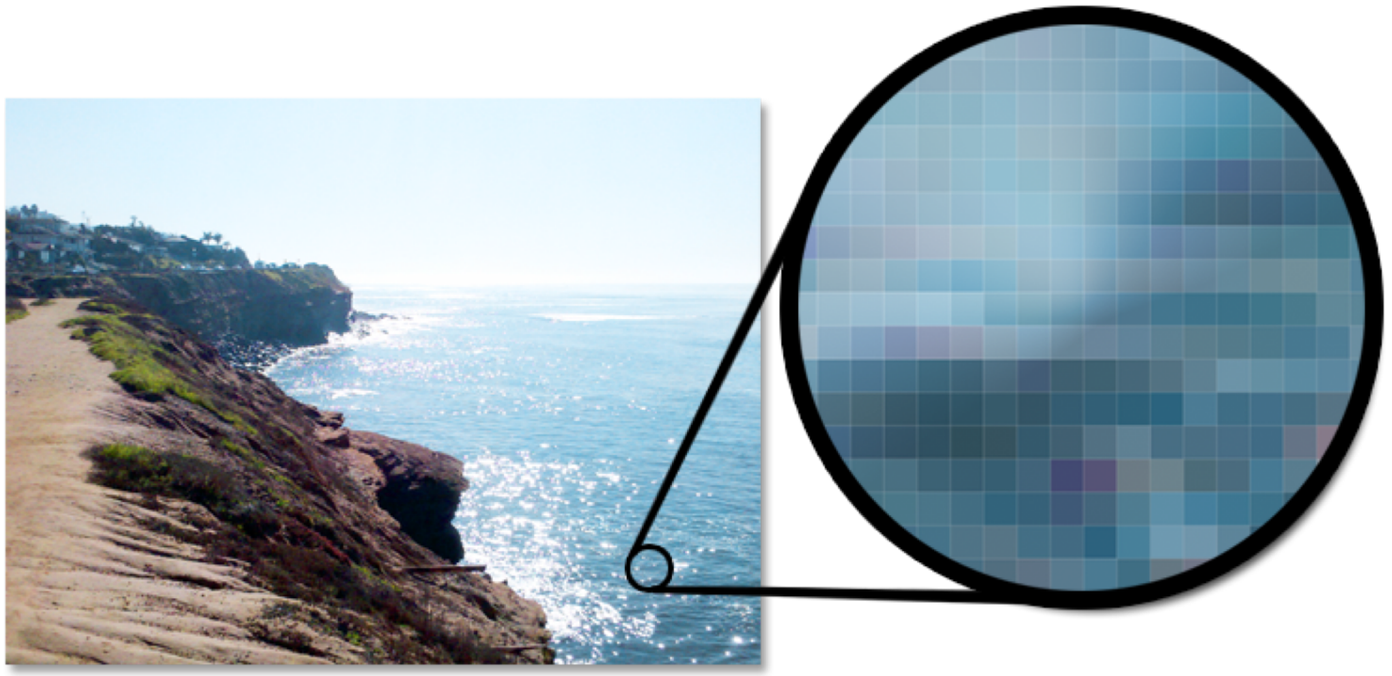
- **Classification** is a machine learning problem for solving a set value given the knowledge we have about that value.

- Many classification problems are trying to predict *binary* values.

- For example, we may be using patient data (medical history) to predict whether the patient is a smoker or not.

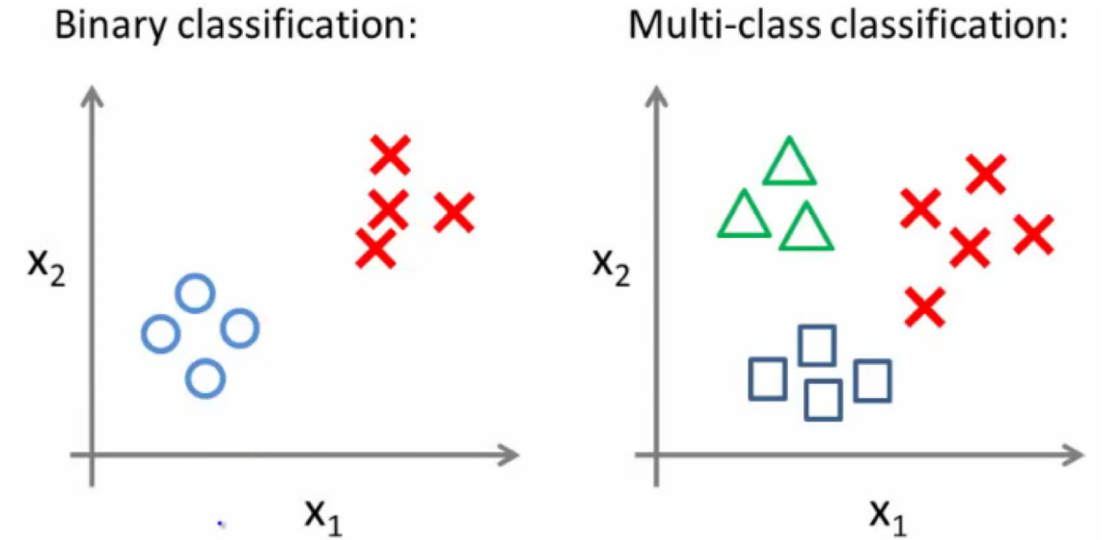
WHAT IS CLASSIFICATION?

- Some problems don't appear to be binary at first glance.
- What if you are predicting whether an image pixel will be `red` or `blue`?
- This is similar to the concept of dummy variables.



WHAT IS CLASSIFICATION?

- Binary classification is the simplest form of classification.
- However, classification problems can have multiple *class labels*.
- Instead of predicting whether the pixel is red or blue, you could predict whether the pixel is red, blue, or green.



WHAT IS A CLASS LABEL?

- A **class label** is a representation of what we are trying to predict: our *target*.

- Examples of class labels from before are:

Data Problem	**Class Labels**
Patient data problem	is smoker, is not smoker
pixel color	red, blue, green

DETERMINING REGRESSION OR CLASSIFICATION

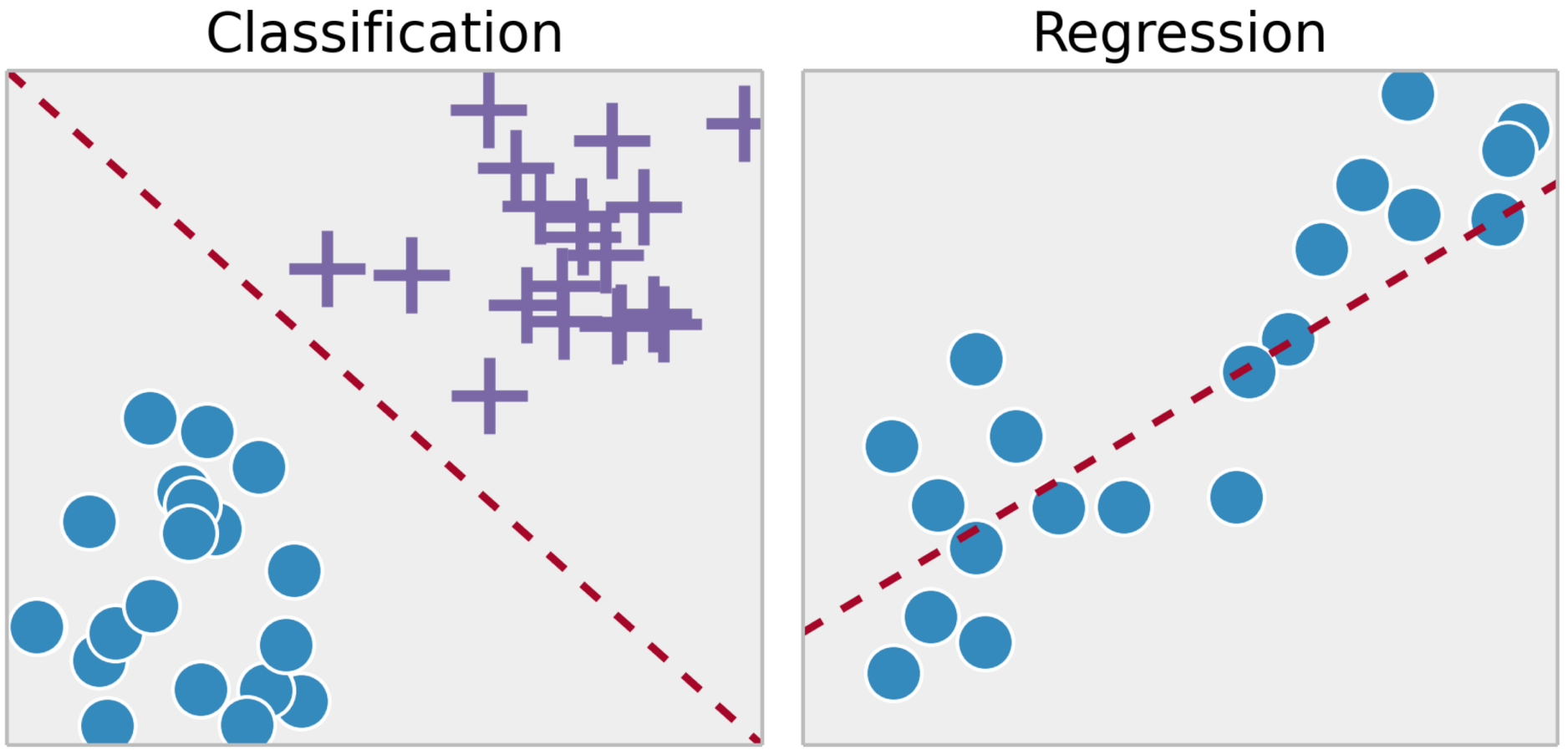
- One of the easiest ways to determine if a problem is regression or classification is to determine if our *target* variable can be ordered mathematically.

- For example, if predicting company revenue, `$100MM` is greater than `$90MM`. This is a *regression* problem because the target can be ordered.

- However, if predicting pixel color, `red` is not inherently greater than `blue`. Therefore, this is a *classification* problem.

DETERMINING REGRESSION OR CLASSIFICATION

- Classification and regression differ in what you are trying to predict.



GUIDED PRACTICE

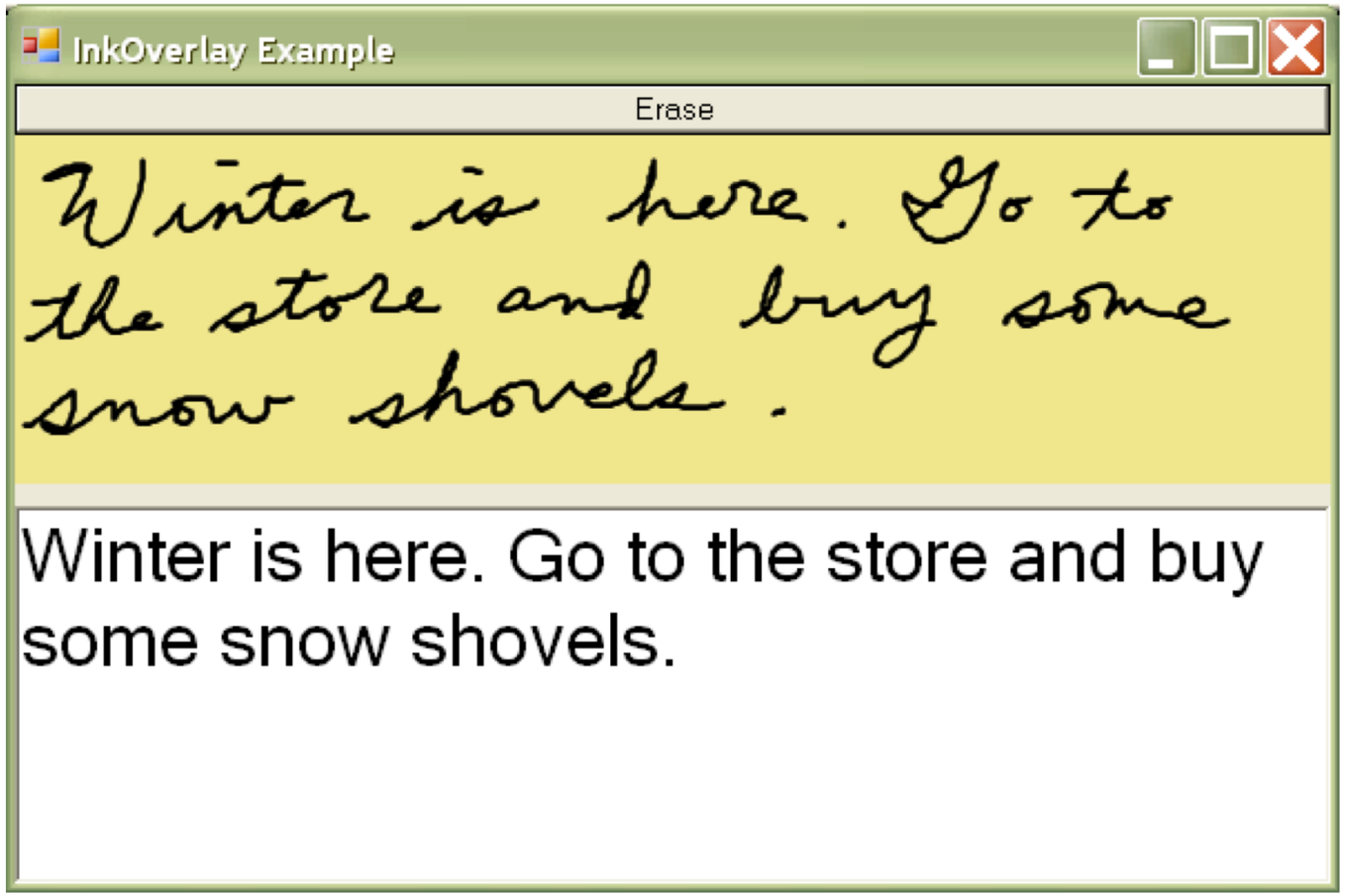
REGRESSION OR CLASSIFICATION?

GUIDED PRACTICE: GUESS THAT METHOD!

GUIDED PRACTICE: GUESS THAT METHOD!

GUIDED PRACTICE: GUESS THAT METHOD!

GUIDED PRACTICE: GUESS THAT METHOD!



ACTIVITY: REGRESSION OR CLASSIFICATION?

EXERCISE

DIRECTIONS

Review the following situations and decide if each one is a linear regression problem, a binary classification problem, a or multi-label classification problem, or neither:

1. Using the total number of explosions in a movie, predict if the movie is by JJ Abrams or Michael Bay.
2. Determine how many tickets will be sold to a concert given who is performing, where, and the date and time.
3. Using anonymized tweets, identify the sender
4. Using data from four cell phone microphones, reduce the noisy sounds so the voice is crystal clear to the receiving phone.
5. With customer data, determine if a user will return or not in the next 7 days to an e-commerce website.

DELIVERABLE

Answers to the above questions

INDEPENDENT PRACTICE

BUILD A CLASSIFIER!

ACTIVITY: BUILD A CLASSIFIER!

DIRECTIONS (20 minutes)

1. Re-explore the iris dataset and build a program that classifies each data point. Use if-else statements and some Pandas functions.
2. Measure the *accuracy* of your classifier using the math of "total correct" over "total samples".
3. Your classifier should be able to:
   a. Get one class label 100% correct (one type of iris is easily distinguishable from the other two).
   b. Accurately predict the majority of the other two classes with some error (hint: make sure you *generalize*).

DELIVERABLE

Classification program for the iris dataset

ACTIVITY: BUILD A CLASSIFIER!

EXERCISE

STARTER CODE

```
from sklearn import datasets, neighbors, metrics
import pandas as pd

iris = datasets.load_iris()
irisdf = pd.DataFrame(iris.data,
columns=iris.feature_names)
irisdf['target'] = iris.target
cmap = {'0': 'r', '1': 'g', '2': 'b' }
irisdf['ctarget'] = irisdf.target.apply(lambda x:
cmap[str(x)])
```

ACTIVITY: BUILD A CLASSIFIER!

EXERCISE

STARTER CODE

```
irisdf.plot('petal length (cm)', 'petal width (cm)’,
      kind='scatter', c=irisdf.ctarget)
irisdf.describe()
```

ACTIVITY: BUILD A CLASSIFIER!

DIRECTIONS

Answer the following questions.

1. How simple could the if-else classifier be while remaining *relatively* accurate?
2. How complicated could our if-else classifier be and remain *completely* accurate? How many if-else statements would you need, or nested if-else statements, in order to get the classifier 100% accurate? (The above uses a count of 2).
3. Which if-else classifier would work better against iris data that it hasn't seen? Why is that the case?

DELIVERABLE

Answers to the above questions

CLASSIFICATION METRICS

INTRODUCTION TO CLASSIFICATION METRICS

- Classification is a different problem from regression
- It therefore requires different metrics
- We are still interested in the wrongness of predictions, and making them less-wrong

INTRODUCTION TO CLASSIFICATION METRICS

- We'll use two primary metrics: *accuracy* and *misclassification rate*.
- **Accuracy** is the number of *correct* predictions out of all predictions in the sample. This is a value we want to *maximize*.
- **Misclassification rate** is the number of *incorrect* predictions out of all predictions in the sample. This is a value we want to *minimize*.
- These two metrics are directly opposite of each other.
- 1 - misclassification rate = accuracy

INTRODUCTION TO CLASSIFICATION METRICS

- **WARNING**: You cannot use regression evaluation metrics for a classification problem, or vice versa. This is a common mistake.

- sklearn will not intuitively understand if you are doing regression or classification, so make sure to manually review your metrics.

WHAT IS K NEAREST NEIGHBORS?

WHAT IS K NEAREST NEIGHBORS?

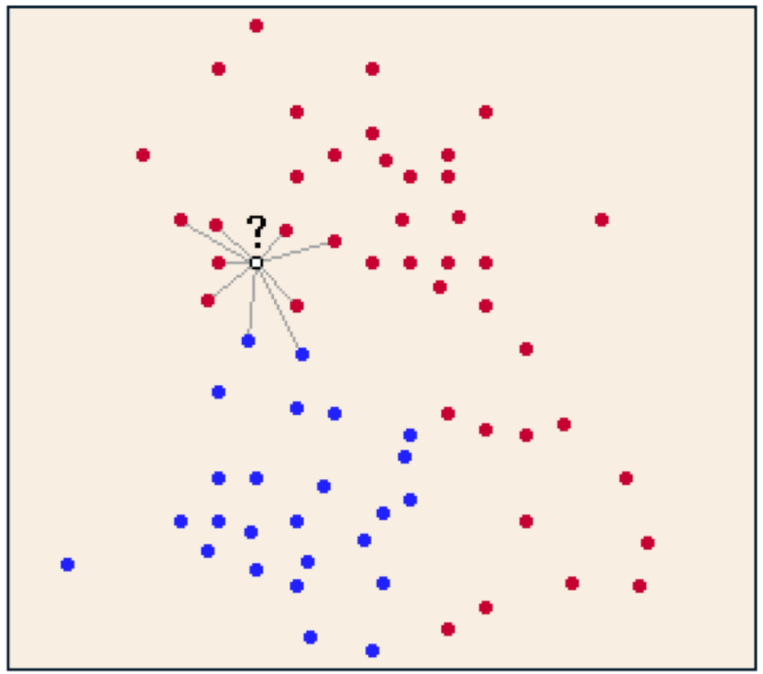
- **K Nearest Neighbors (KNN)** is a classification algorithm that makes a prediction based upon the closest data points.

- The KNN algorithm:
  - For a given point, calculate the distance to all other points.
  - Given those distances, pick the *k* closest points.
  - Calculate the probability of each class label given those points.
  - The original point is classified as the class label with the largest probability (“votes”).

WHAT IS K NEAREST NEIGHBORS?

- KNN uses distance to predict a class label. This application of distance is used as a measure of similarity between classifications.

- We’re using shared traits to identify the most likely class label.



WHAT IS K NEAREST NEIGHBORS?

- Suppose we want to determine your favorite type of music. How might we determine this without directly asking you?

- Generally, friends share similar traits and interests (e.g. music, sports teams, hobbies, etc). We could ask your five closest friends what their favorite type of music is and take the majority vote.

- This is the idea behind KNN: we look for things similar to (or close to) our new observation and identify shared traits. We can use this information to make an educated guess about a trait of our new observation.

ACTIVITY: KNOWLEDGE CHECK

ANSWER THE FOLLOWING QUESTIONS

1. In what other tasks do we use a heuristic similar to K Nearest Neighbors?

DELIVERABLE

Answers to the above questions

LET'S CODE